


Nr. 7/85 3. Jahrgang 5,50 DM 16,50 FMK 6,00 sFr 48 öS

# Computing Programming Using



**Aktuell:** Die „Neuen" von Atari u. Commodore

Der Epson Personal-Computer QX-16 (Bericht Seite 78)




## Software-Tests:
CPC-464: Forest at World's End
CPC-464: Ghouls

## User:
TI-99: Super-Disassembler
CPC-464: PAINTER
BTX: – Aktuell –

## Tips und Tricks:
Für CPC-464

## Utility:
Atari: Zeicheneditor
C-64 / VC-20: Decelerator
CPU-Game
MSX: Alien Attack

C 64 * CPC 464 * VC-20 * Atari * Apple * TI-99 * ZX-Spectrum * MSX *

**28 Computersysteme im Benchmark-Test!**

# Ab sofort wieder bei Ihrem Zeitschriftenhändler erhältlich!




„COMPUTE MIT..." haben wir, lieber Leser, speziell auf die Commodore- und Schneider- (hier der CPC 464)-Computer zugeschnitten. Nachdem sich schon im Herbst 1984 auf dem deutschen Computer-Markt abzuzeichnen schien, daß der CPC 464 nach einem nahezu kometenhaften Aufstieg gute Chancen besitzt, dem Giganten Commodore ein ernstzunehmender Konkurrent zu werden, wurde unser Team nahezu gezwungen, diesen Trend aufzunehmen und auch weiterhin zu verfolgen. D. h., für beide populären Homecomputer-Riesen werden wir in jeder Ausgabe von „COMPUTE MIT..." ein attraktives Angebot bereithalten. Außerdem bietet unsere Redaktion einen Super-Service an, um Ihnen eventuelle schwierige Abtippereien zu ersparen. Mehr darüber auf Seite 11!




Report – Werkstatt – Hardware – Bücher – Software – Tips & Tricks – Assemblerkurs Teil 1 – Jungle – Sequenzer – Olympia – Cobra – Krümi – Q-Bert u. v. mehr

**STOP +++ Neue Ausgabe +++ STOP**

# CPU Juli '85


Der OKI MICROLINE 192-Drucker, eine der zahlreichen Neuheiten, die seit der Hannover-Messe auf den Markt gekommen sind


Ein komfortables Zeichenprogramm für den CPC-464



Ein Weltraumspiel für MSX-Computer-Besitzer

## CPU-Utility



## CPU-User



## CPU-Tips und Tricks



## CPU-Markt



## CPU-Game



## CPU-Test



## Rubriken

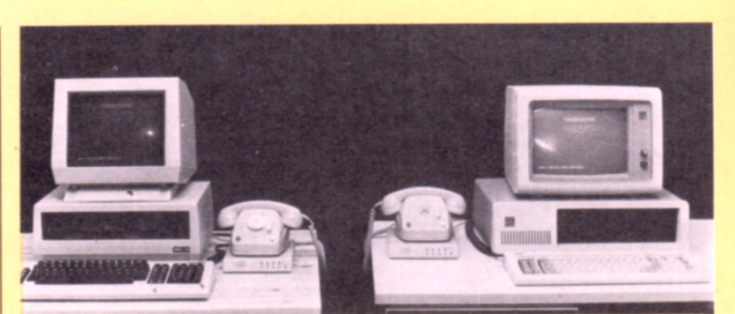
# Erstes vollautomatisches Selbstwähl-Modem von der Deutschen Bundespost FTZ-zugelassen

Unter der Nummer FTZ 04168 D wurde erstmals von der Deutschen Bundespost ein Modem zugelassen, das auf der Basis einer Wählelektronik DB 1201/02 arbeitet und von einem Programm gesteuert wird.
Damit besteht jetzt eine kostengünstige, sichere Alternative zu den störanfälligen Akustikkopplern, zu teuren Standleistungen und z. T. zum Telex-/Teletextnetz.
Das Wählmodem wurde zugelassen für alle IBM PC-Modelle, Sirius/Victor-Computer und alle IBM-Kompatiblen.
Das Modem wird asynchron mit 1200/75 Baud betrieben und mit einem von Controlware entwickelten Gehäuse mit serieller Schnittstelle und Spannungsversorgung an den PC angeschlossen.

**Die midas-Software arbeitet in 2 Modi:**

**manuell:**
Telefonnummern werden über die Tastatur eingegeben, der PC wählt sich zu einem weiteren durch. Wenn die Verbindung zustandekommt, können Dateien gesendet oder von der Gegenanlage abgeholt werden.

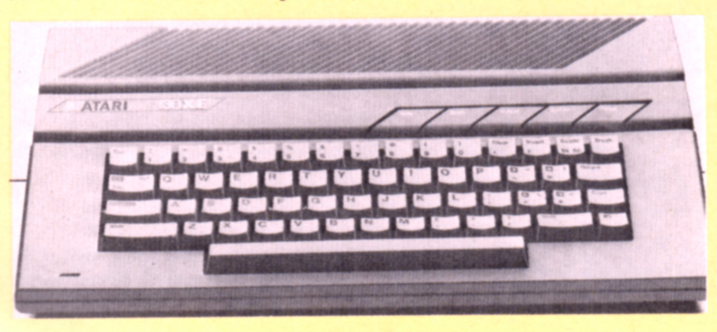
**automatisch:**
Aus dem midas-Programm oder einem beliebigen anderen kann eine Datei erstellt werden, die beliebig viele Telefonnummern, gewünschte Uhrzeit und zu übertragende Texte beinhaltet. Diese Datei wird vom Programm abgearbeitet und die Texte können z. B. nachts übertragen werden.



Übertragen wird 8-bit-Code, also nicht nur reiner Text, wie im Telex-Netz, sondern auch Maschinenprogramme oder Grafikdateien.
Die Übertragung erfolgt komprimiert. Für mehr als 3 gleiche Zeichen werden nur das Zeichen selbst und dessen Anzahl übertragen. Damit können die Kosten wesentlich reduziert werden.
Die Wählkarte DB 1201/02 wird von der Deutschen Bundespost für 18 DM/Monat vermietet. Software und Modem-Box sind von midas für den Preis von DM 1500 + 14 % MwSt. erhältlich.
In Kürze wird das System auch im Hintergrundbetrieb verfügbar sein. Damit kann der PC senden/empfangen, wenn im Vordergrund z. B. Briefe geschrieben werden.

# Neue Computer: Ataris neue Superrechner

Auf der CES in Las Vegas stellte Atari erstmals seine neue Rechnergeneration dem Publikum und der Fachpresse vor. Auf der Hannover-Messe wurden die Geräte dem staunenden Publikum präsentiert:
Das neue Flaggschiff von Jack Tramiel ist der Atari ST 520. Dieser neue Computer ist mit einer 8 MHZ – 68000 CPU und 512 kByte RAM ausgerüstet. Aufgrund des Graphic Enviroment Managers GEM von Digital Research, einem Graphik-„Betriebssystem“, bietet der ST 520 nahezu die gleiche Benutzeroberfläche wie Apples Macintosh. GEM unterstützt Icons, Windows und die Atari-Mouse. Aufgrund dieser Mac-Ähnlichkeit trägt der neue Rechner bereits den Spitznamen Jackintosh. Außer dem GEM ist in den 192 kByte ROM das Betriebssystem TOS (Tramiel Operating System) untergebracht, das eine modifizierte CP/M 68k-Version ist.




Wie auf der Messe zu erfahren war, wird der ST 520 ab Juni/Juli 85 mit Basic und Logo im ROM ausgeliefert, andere Sprachen (wie z. B. C und Forth) werden jedoch auch erhältlich sein. Das beste ist jedoch der Preis des Rechners. Zusammen mit einer ebenfalls neu entwickelten 500 kByte Floppy (Format 3½ Zoll), einem monochromen Monitor und der Maus wird das Komplettsystem nach Angaben von Atari-Mitarbeitern etwa 2990 DM kosten. Zum Sensationspreis von nur ca. 1200 DM soll eine 10 MB-Festplatte erhältlich sein.
Die auf der CES ebenfalls angekündigten kleineren Modelle der ST-Serie (ST 130 und 260) werden nach Angaben von Atari-Mitarbeitern zumindes vorerst nicht in Europa angeboten.
Zusammen mit dem ST 520 stellte Atari auch den Nachfolger des 800 XL, den 130 XE vor. Das neue Gerät hat

einen RAM-Speicher von 128 kByte, eine 6502-CPU und soll vollkompatibel zu seinem Vorgänger sein. Seine kleineren Brüder, die 64 kByte-Modelle 65 XE, 65 XEP und 65 XEM wurden nicht mehr vorgestellt und werden wohl auch (zumindest vorerst) nicht produziert.

# Atari ST – ein neuer Leistungs-PC –

Bisher konnte der Kunde nur zwischen zwei Computer-Klassen wählen. Preiswerte, dafür aber einfache Heim-Computer mit wenig Leistung oder leistungsfähige, dafür aber sehr teure Personal-Computer. Dazwischen in der Klasse zwischen 3000 und 6000 Mark gab es kein besonders attraktives Angebot. Atari hat jetzt für diesen Käuferkreis einen Personal-Computer der Spitzenklasse zu einem sehr günstigen Preis gebaut. Mit 520 Kilobyte-Arbeitsspeicher und einem 500 Kilobyte-Diskettenlaufwerk wird der Atari unter 3000 Mark kosten.



Die von Atari beim ST eingesetzte Computer-Technologie repräsentiert die Spitze des heute möglichen Standards. Dazu gehört besonders der Mikroprozessor MC 68000, der zu den schnellsten und leistungsfähigsten Prozessoren überhaupt gehört. Auch umfangreiche Aufgaben und große Datenmengen im professionellen Einsatz sind für den 520 ST kein Problem.

Mit GEM wird ein völlig neues Betriebs- und Bedienungssystem für Personal-Computer angeboten. Es wurde vom international bekannten Software-Haus Digital Research entwickelt. GEM bietet dem Benutzer den großen Vorteil, daß er keine komplizierten Computer-Befehle mehr lernen muß, sondern im einfachen Dialog auf verständliche Fragen des Computers antwortet.

Dafür steht ihm neben der Tastatur, natürlich mit Umlauten und deutschen Sonderzeichen, eine sogenannte Maus zur Verfügung, die einfach über den Tisch geschoben wird. Will er beispielsweise schnell etwas berechnen, wird der Pfeil mit Hilfe der Maus auf dem Symbol „Kalkulator“ bewegt. Nach einem Knopfdruck erscheint die Zeichnung eines Taschenrechners auf dem Monitor, und wie mit dem Finger werden dann die zu berechnenden Zahlen mit der Maus und dem Pfeil angetippt.

Software-Entwickler in den USA und der Bundesrepublik arbeiten an Standard- und Branchenprogrammen für den Atari ST.

Die ST-Modelle von Atari sind komplett ausgestattet. Ohne elektronische Aufrüstung oder Zusatzplatinen sind zwei Schnittstellen, RS232C und Centronics, vorhanden. Außerdem ein Anschluß für das serienmäßige 3,5 Inch Diskettenlaufwerk und eine Festplatte. Atari hat Steuerung und Intelligenz für Floppy und Festplatte in diese Systeme integriert, so daß sie ohne Aufwand steckfertig an den Kunden geliefert werden. Für den Bildschirm gibt es drei Anschlüsse, RGB für einen Farb-Monitor, Farbe ist serienmäßig, eine AV-Buchse für einen monochromen Monitor und ein HF-Ausgang für den Anschluß eines normalen Fernsehgerätes. Durch die Auswahl unter drei Bildschirmen gewinnt der Atari 520 ST an Flexibilität. Bisher war der Personal-Computer wegen seiner Größe und seinem Gewicht an den Arbeitsplatz gebunden. Der kleine, leichte, handliche ST dagegen ist mobil einsetzbar.

Preis des Atari 520 ST inklusive eines 3,5-Zoll-Diskettenlaufwerks: DM 2795,–

# Die neuen 128-Modelle von Commodore

Der neue C 128-Personal-Computer deckt alle Anwendungen bisheriger Heimcomputer ab, erfüllt darüber hinaus noch die Forderungen an einen Personal-Computer. Das Modell folgt damit dem Trend bei den häuslichen Anwendungen. Traten bisher im wesentlichen Jugendliche als Benutzer auf, so werden im zunehmendem Maße auch Erwachsene für Aufgaben der beruflichen Fortbildung, Einsatz im Kleinstbetrieb oder Teilnahme an Service-Einrichtungen der Telekommunikation (Btx, Datex-P, Mailbox, Datenbanken, Electronic Mail) hinzukommen.

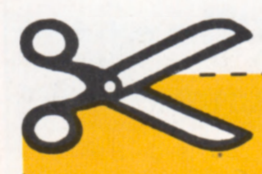
Diesem Anwendungsprofil entsprechend ist die Struktur des Commodore 128 als Dual CPU-System mit MOS 8502 und Z 80 aufgebaut. Die neuentwickelte 8502 in Verbindung mit residentem Betriebssystem und Basic-Interpreter ist im Commodore 64-Modus zu diesem Commodore-Modell kompatibel; im Commodore 128-Modus arbeitet sie kompatibel zu der am weitesten verbreiteten 8-bit-CPU 6502. Hierdurch sind praktisch sämtliche für Commodore-Rechner jemals entwickelte Programme nach geringen Modifikationen lauffähig. Hinzu kommt das

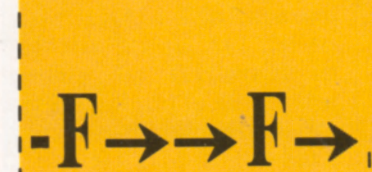
Angebot an professioneller CP/M 3.0 Software, die über Disketten geladen wird. Der Arbeitsspeicher beträgt je nach Betriebsmodus 64 bis 128 KB RAM. Die Kapazität läßt sich bis auf 512 KB erweitern. Im Textbetrieb ist der Bildschirm in 40 Zeichen x 25 Zeilen organisiert. Im 128- und CP/M-Modus kann auf 80 Zeichen umgeschaltet werden. Gleiches gilt für die Grafikauflösung mit 320 bzw. 640 x 200 Punkten. Es werden bis zu 16 Farben dargestellt, im 64- und 128-Modus bis zu 8 Sprites. Das Schnittstellenangebot ist wie gewohnt komplett, jedoch gibt es zusätzlich neben dem Video-Ausgang einen RGBI-Ausgang. Das in der Modellversion Commodore 128 D eingebaute Diskettenlaufwerk ist kompatibel zur Floppy 1541 und erreicht je nach Betriebsart eine Kapazität von maximal 410 KB. Die alphanumerische Tastatur mit separatem Rechenblock und Funktionstasten hat mit insgesamt 92 Tasten professionellen Zuschnitt. Der Preis für den C 128-Personal-Computer beträgt ca. DM 1200,–. Mit dem neuen Modell C 128 deckt Commodore über das eigene residente Betriebssystem besonders benutzerfreundliche Anwendungen in Unterhaltung und Ausbildung ab und ermöglicht gleichzeitig über CP/M die Nutzung großer Programm-Bibliotheken.



# Technische Daten des C-128

| | |
|---|---|
| CPU: | MOS 8502<br>(simuliert MOS 6510 und MOS 6502)<br>Zilog Z 80 |
| Taktfrequenz: | 64-Modus 1 MHz<br>128-Modus 2 MHz<br>Z80-Modus 4 MHz |
| Arbeitsspeicher: | 64-Modus – 64 KB RAM<br>128-Modus – 128 KB RAM<br>Z80-Modus – 128 KB RAM<br>Externe RAM-Disk bis 512 KB |
| Betriebssystem: | im 128-Modus resident<br>48 KB ROM (max. 96 KB ROM)<br>im Z 80-Modus CP/M 3.0 über Diskette ladbar |
| Interfaces: | Commodore seriell<br>Control-Ports, 2<br>User-Port<br>Audio/Video<br>RGBI, digital (TTL-Pegel)<br>HF-Modulator<br>Erweiterungs-Port für Steckmodule |
| Programmiersprache: | im 64-Modus – BASIC 2.0 residenter Interpreter<br>im 128-Modus – BASIC 7.0 residenter Interpreter und Maschinensprache-Monitor<br>im Z 80-Modus – weitere Sprachen ladbar |
| Massenspeicher: | Typ Commodore 1541, beim Modell Commodore 128 D integriertes 5 ¼-Zoll-Floppy-Laufwerk, sonst separat<br>im 64/128-Modus kompatibel zum Floppy-Laufwerk 1541<br>im-64 Modus – Double Side max. 350 KB, Übertragungsrate 300 Zeichen/sec<br>im 128-Modus – Double Side max. 350 KB, Übertragungsrate 1500 Zeichen/sec<br>im Z 80-Modus – Double Side/Double Density max. 410 KB, Übertragungsrate 3500 Zeichen/sec |
| Tastatur: | professionelle Schreibmaschinentastatur mit 92 Tasten, separater Rechenblock mit 14 Tasten, 8 Funktionstasten, 6 Cursortasten, 1 Help-Taste und 1 Umschalttaste von 40 auf 80 Zeichen |
| Zeichen/Zeile: | Textbetrieb<br>– im 64-Modus 40 Zeichen x 25 Zeilen<br>– im 128-Modus umschaltbar 40/80 Zeichen x 25 Zeilen<br>– im Z 80-Modus 640 x 200 Punkte<br>Graphik<br>- im 64-Modus 320 x 200 Punkte<br>- im 128-Modus umschaltbar 320/640 x 200 Punkte<br>- Z 80-Modus 640 x 200 Punkte |
| Farbe: | 16 Farben<br>im 64/128-Modus zusätzlich bis zu 8 Sprites |
| Sound: | 6581 SID-Chip |
| Monitor: | 40 Zeichen<br>– im 64/128-Modus über VIC Chip NTSC/PAL Color TV oder Monitor 1701/1702<br>80 Zeichen<br>– im 128-Modus und Z 80-Modus über Video Interface Chip 7563 RGBI Monitor plus monchromen Monitor |





## Fortsetzung des CPU-Wörterbuches

### -F→→F→

**Formatierer**
Ein Systemprogramm, meist ein Unterprogramm eines Diskettenbetriebs-Systems, (z. B. CP/M), das die einzelnen Sektoren/Tracks einer Diskette kennzeichnet.

**Forth**
Eine höhere Programmiersprache, die sehr stark an Assembler angelehnt ist. Es ist eine sehr schnelle Sprache, die auch für Heimcomputer geeignet ist. Sie wurde Mitte der siebziger Jahre von der amerikanischen Firma Forth entwickelt und war in erster Linie für Steuerungsaufgaben gedacht. Der Anwender kann in Forth eigene Befehle definieren. Forth läßt sich somit für fast alle Aufgaben einsetzen.

**Funktionstaste**
Tasten eines Computers, denen kein Zeichen, sondern eine Funktion zugeordnet ist; dazu zählen die Cursor-Tasten, Break-Taste und andere Spezialfunktionen des Computers. Viele Computer besitzen die Möglichkeit, bestimmte Tasten mit einer Funktion zu belegen.

**Friktionsantrieb**
Der Friktionsantrieb erlaubt bei Druckern die Verwendung von Endlos- oder auch Einzelblattpapier ohne den „Lochrand", der für die Traktorführung erforderlich ist.

**Fortran**
Eine höhere Programmiersprache, die im Jahre 1954 entwickelt wurde. Ähnlich wie Algol ist auch Fortran eine problemorientierte Programmiersprache, die vor allem im technisch-wissenschaftlichen Bereich eingesetzt wird. Die Symbolik von Fortran lehnt sich stark an die mathematische Formelsprache an und ist daher direkt auf die Bedürfnisse von Ingenieuren, Technikern und Wissenschaftlern zugeschnitten.

### -G→→G→

**Ganzzahl** (integer Zahl)
Eine Zahl, die keine Nachkommastellen besitzt. Viele Heimcomputer können spezielle Integer-Variablen verarbeiten, diese benötigen wesentlich weniger Speicherplatz als die Realen-Variablen.

**Gleitkomma-Arithmetik**
Häufig auch mit Fließpunkt-, oder Floating-Point-Arithmetik bezeichneter 8 Rechenalgorithmus.

**GOTO**
Ebenfalls ein Befehl der Programmiersprache Basic, welcher zu der angegebenen Zeilennummer verzweigt.

**GOSUB**
Befehl der Programmiersprache Basic. Durch die Angabe einer Zeilennummer wird ein Unterprogramm aufgerufen. Dieses wird durch den RETURN-Befehl wieder verlassen.

### -H→→H→

**Halbduplex**
Mit Halbduplex bezeichnet man ein Datenübertragungsverfahren, bei dem zwar die Übertragung in beide Richtungen möglich ist, jedoch zum selben Zeitpunkt Daten nur in eine Richtung übertragen werden können. Um eine Antwort zu senden, müssen Sender und Empfänger umgeschaltet werden. Halbduplex wird heute kaum noch verwendet und wurde durch das Vollduplex-Verfahren ersetzt.

**Haltepunkt (breakpoint)**
In einem Programm gekennzeichnete Stelle, an der dieses angehalten wird und dem Benutzer die Möglichkeit zum Testen bietet. Diese Methode wird vorwiegend in der Assembler-Programmierung angewandt.

**Handler (hantierer)**
Teil eines Programms, um Peripheriegeräte zu kontrollieren.

**Hamming-Code**
Ein spezieller Code, der als Fehlerkorrektur-Einrichtung verwendet wird. Es ist ein 7-Bit-Code, dessen Aufbau die Identifizierung fehlerhaft übertragener Daten ermöglicht.

**Hardcopy**
Ein genauer Ausdruck einer Bildschirmseite. Oft ist der Hardcopy-Befehl im ROM des jeweiligen Computers enthalten.

**Hardware**
Alle Geräte und mechanischen Komponenten eines Computersystems einschließlich der Peripheriegeräte.

**HEX**
Abkürzung für Hexadezimalsystem.

**Hinterbandkontrolle**
Bei der Hinterbandkontrolle werden die gerade aufgezeichneten Daten mit einem Lesekopf, der gleich hinter dem Schreibkopf sitzt, geprüft und mit den ursprünglichen Sollwerten verglichen. Tritt bei der Prüfung ein Fehler auf, so werden die Daten neu auf das Band geschrieben. Diese komfortable Einrichtung ist bis jetzt bei Heimcomputern noch nicht vorhanden.

# „Quassi“: Ein Programm für den ZX-Spectrum 48 K

Quassi ist ein Programm, das die Erstellung und Verbesserung von Maschinenprogrammen wesentlich komfortabler gestaltet. Es gestattet die Eingabe des Codes im Hex-Format, setzt selbständig die entsprechenden Speicherzellen, wie man es von einem Assembler gewohnt ist, auf den Bildschirm und erleichtert das Verbessern oder Löschen einzelner oder mehrerer Speicherzellen. Nach dem Start erscheinen zunächst die verschiedenen möglichen Kommandos. Danach gibt man die Anzahl der Bytes ein, die man sich reservieren lassen will, und anschließend die Adresse, ab der man das Programm erstellen und starten lassen will. Es erscheint darauf die erste Adresse mit einem Cursor davor. Man kann jetzt den gewünschten Hex-Code oder das Kommando eingeben. Ist es ein Hex-Code, so wird dieser der aktuellen Adresse zugeordnet. Dabei ist es vorgesehen und auch sinnvoll, jeweils eine geschlossene Befehlsgruppe in einer Zelle unterzubringen. Dies erhöht die Übersichtlichkeit beträchtlich. Nach Übergabe des Hex-Codes mit ENTER erscheint dieser nun neben seiner Adresse, es wird die nächste freie Adresse mit Cursor angezeigt und man kann einen weiteren Code oder ein Kommando eingeben.

**Zu den Kommandos:**

"p" druckt das gesamte, bisher erstellte Programm aus,
"l" lädt das Programm ab Startadresse in den Speicher,
"g" startet das gespeicherte Programm ab Startadresse,
"s" speichert das Programm auf Kassette,

"x a, b" löscht von Adresse a an b Zeilen. Dabei werden jeweils ganze Zeilen und nicht nur Speicherzellen gelöscht.

```
Aus 65200 cd 1f 09 ##
    65203 f1 ##
    65204 06 00 ##
    65206 0e ff ##

wird mit "x65203,2": 65200 cd 1f 09##
                     65203 0e ff ##
```

"v a, b$" Verbessert Adresse a mit b$:

```
65200 cd 00 91 ## → "v65201,10" → 65200 cd 10 91 ##
```

"i a, b$" Schiebt ab a noch b$ ein:

```
65200 0e 00 ##    }                     { 65200 0e 00 ##
                  } "i65202,0b6e" →     { 65202 0b 6e ##
65202 cd 3e 09 ## }                     { 65204 cd 3e 09 ##
```

"r a, b" Berechnet einen relativen Sprung von a nach b. Dabei ist a die Adresse, in der der relative Sprungbefehl steht. Für die Sprungweite setzt man am besten vorläufig einen beliebigen Wert ein und läßt am Schluß den richtigen Wert ausrechnen und einsetzen:

```
65200 4f ##     }                       { 65200 4f ##
65201 18 ff ##  } →"r65201,65200"→       { 65201 18 fd ##
65203 00 ##     }                       { 65203 00 ##
```

"xx" löscht das ganze Programm, setzt die Adresse wieder auf die Startadresse und geht zum Anfang.

Wenn man die Kommandos eingibt, so muß man darauf achten, daß nach dem 1. Kommandobuchstaben direkt die anzusprechende Speicherzelle steht (natürlich außer p, l, g, s, xx) und daß die zweite Adresse bzw. der HexCode (bei v und i) vom ersten Teil durch Kommas getrennt werden. Beispiele für die Kommandos:

v65200,ff : i65190,0b0a : x64500,1

Nachdem man das Maschinenprogramm erstellt hat, lädt man es mit "l" und speichert es mit "s" am besten gleich ab. Danach kann man es mit "g" laufen lassen. Der Variablen "Ergebnis" wird dabei der Inhalt des BC-Registerpaares zugeordnet.
Der Speicherbereich von 64900 an aufwärts ist mit Absicht freigelassen, um

a) die UDGs nicht zu stören

und b) evt. dort noch ein MC-Programm unterzubringen, das eine Einzelschrittbearbeitung erlaubt. Dazu müßte man jedes "##" durch ein 'CALL-Subroutine' ersetzen, daß dieses Programm aufruft.

# Wichtige Programmteile:

| Zeile | | |
|---|---|---|
| | 40 – 290 | Eingabe des Hex-Codes Verarbeitung des Hex-Codes mit Verzweigung zu den Unterprogrammen |
| | 300 – 305 | Cursor setzen |
| | 310 – 315 | Cursor löschen |
| | 400 – 430 | Unterprogramm “Laden“ |
| | 500 – 510 | Unterprogramm “Starten“ |
| | 600 – 650 | Unterprogramm “Verbessern“ |
| | 700 – 765 | Unterprogramm “Löschen“ |
| | 800 – 840 | Unterprogramm “Einfügen“ |
| | 900 – 935 | Unterprogramm “Relativer Sprung“ |
| | 1100 – 1130 | Unterprogramm “Programm listen“ |
| | 1200 – 1210 | Unterprogramm “Abspeichern“ |
| | 5025 – | Unterprogramm “Bildschirmkopf“ |
| | 5100 – 5120 | Unterprogramm “neue Bildschirmseite“ |
| | 5300 – 5320 | Unterprogramm “alte Bildschirmseite löschen“ |
| | 8000 – 8025 | Unterprogramm “Eingabestring aufspalten“ |
| | 9000 | Unterprogramm “Unterprogramme aufrufen“ |

Sollte das Programm wegen einer fehlerhaften Eingabe aussteigen (ein Komma vergessen o. ä.), so setzt man es am besten mit GOTO 40 wieder fort. Der bis dahin erstellte Programmstring geht dabei nicht verloren.
Gegen Absturz bei einem Programmfehler im erstellten Maschinenprogramm ist natürlich auch Quassi (Quasi-Assembler) nicht gefeiht. Deshalb noch mal der Tip: Erst abspeichern, dann starten. Aber hier könnte man ja im reservierten Speicher oberhalb 64900 noch eine Absturzroutine unterbringen, die das Schlimmste verhindert. Solche Routinen mit ein paar Bytes Länge gibt es genügend.

```
   2 PRINT AT 2,0;"Kommandos:";AT 4,4;"x(a),(b) Loesche von Adresse
     a aus b Zeilen";AT 6,4;"xx        Loesche ganzes Prg.";AT 7,4;"1
    Lade Programm ";AT 8,4;"g        Starte Programm ab
Startadresse"
   3 PRINT AT 10,4;"i(a),b$  Schiebe ab Adresse                a b$ ein";A
T 12,4;"r(a),(b) Berechne relati-              Sprung von a nach b";AT
  14,4;"p        Drucke                        ganzes Programm aus"
   4 PRINT AT 16,4;"v(a),b$  Verbessere Adresse                a mit b$";A
T 18,4;"s        Abspeichern";AT 21,0;"Bitte eine Taste druecken": PAUSE
 0
   5 CLS : PRINT AT 0,0;"Ram frei bis 64900"
  10 INPUT "Wieviel Bytes fuer das Programm reservieren? ";ram
  15 LET ramt=64900-ram: CLEAR ramt-1
  17 LET ramt=PEEK 23730+256*PEEK 23731
  19 PRINT AT 0,0;"Ramtop:                 ";ramt;
  20 INPUT "Startadresse? ";start: IF start<=ramt OR start>64900 THEN  G
O TO 20
  21 GO SUB 5025
  22 LET zz=0: LET prog=0: LET z=0: LET pos=0: LET h$="": DIM v(18)
  25 PRINT AT 3,2;"Adr   Hex-Inhalt"
  30 GO SUB 300
  31 LET adr=start
  32 LET v(1)=500: LET v(6)=400
  33 LET v(16)=600: LET v(3)=800
  34 LET v(10)=1100: LET v(12)=900
  35 LET v(13)=1200: LET v(14)=1500
  36 LET v(18)=700
  40 PRINT AT pos+5,1;adr
  42 INPUT "Hex-Eingabe: ";i$
  47 IF CODE i$(1)<48 OR CODE i$(1)>122 THEN  GO TO 40
  50 LET l=0: LET b=LEN i$
  55 LET l=l+1
  57 IF l>b THEN  GO TO 70
  60 IF i$(l)<>" " THEN  GO TO 55
  65 LET i$=i$( TO l-1)+i$(l+1 TO ): GO TO 57
  70 IF CODE i$(1)>102 THEN  GO TO 9000
  72 IF b/2<>INT (b/2) THEN  GO TO 40
  75 LET i$=i$+"##"
```

```
  80 LET h$=h$( TO 2*z)+i$+h$(2*z+1 TO )
  90 GO SUB 310
  95 LET x$=""
 100 FOR g=1 TO LEN i$/2
 105 LET x$=x$+i$(2*g-1 TO 2*g)+" "
 110 LET z=z+1: NEXT g
 195 PRINT AT pos+5,7;x$
 200 LET pos=pos+1
 210 LET adr=adr+g-2
 250 IF adr<64900 THEN  GO TO 290
 255 PRINT : PRINT "  Adresse zu gross, bitte neue    Startadresse einge
ben.Nach       Laden des Programms mit CLEAR   neuen Ramtop setzen!"
 260 INPUT "Neue Startadresse: ";start
 265 LET adr=start: GO SUB 5025: GO SUB 1100
 290 GO SUB 300: GO TO 40
 300 IF pos>=16 THEN  LET zz=0: GO SUB 5100
 305 PRINT AT pos+5,0; INK 2; FLASH 1;">": RETURN
 310 PRINT AT pos+5,0;" "
 315 RETURN
 400 LET h$=h$+"c9##": LET z=z+2
 405 LET sp=start-1: LET m=-1
 410 LET sp=sp+1
 415 LET m=m+2: IF m>LEN h$ THEN  LET prog=1: RETURN
 420 IF h$(m)="#" THEN  GO TO 415
 425 POKE sp,16*(CODE h$(m)-48-39*(h$(m)>"9"))+CODE h$(m+1)-48-39*(h$(m+
1)>"9")
 430 GO TO 410
 500 IF prog=0 THEN  PRINT AT 21,9;"Kein Programm!": PAUSE 50: RETURN
 505 CLS : LET ergebnis=USR start
 510 GO SUB 5025: RETURN
 600 IF LEN i$<2 THEN  RETURN
 605 GO SUB 8000
 620 IF ad<0 OR ad+start>=adr THEN  RETURN
 625 LET i=1: LET a=-1
 630 IF h$(2*i)<>"#" THEN  LET a=a+1
 635 IF a=ad THEN  LET h$=h$( TO 2*i-2)+i$( TO 2)+h$(2*i+1 TO ): GO TO 6
45
 640 LET i=i+1: GO TO 630
 645 LET zz=1: GO SUB 1100
 650 RETURN
 700 IF LEN i$<2 THEN  RETURN
 705 IF i$(2)="x" THEN  GO SUB 5305: LET adr=start: GO TO 22
 710 GO SUB 8000: IF ad<0 OR ad+start>=adr THEN  RETURN
 715 LET gg=1: LET du=VAL i$(1)
 717 LET i=1: LET a=-1
 720 IF h$(2*i)<>"#" THEN  LET a=a+1
 725 IF a=ad THEN  LET cc=i: GO TO 740
 730 LET i=i+1: GO TO 720
 740 LET i=i+1
 745 IF h$(2*i)="#" THEN  LET h$=h$( TO 2*cc-2)+h$(2*i+1 TO ): GO TO 755

 750 GO TO 740
 755 LET z=z-i+cc-1
 760 IF gg<du THEN  LET gg=gg+1: GO TO 717
 765 IF h$(LEN h$)<>"#" THEN  LET h$=h$+"##"
 770 LET zz=1: GO SUB 1100: RETURN
 800 IF LEN i$<2 THEN  RETURN
```

```
805 GO SUB 8000
807 IF ad<0 OR ad+start>=adr THEN  RETURN
810 LET i=1: LET a=-1
815 IF h$(2*i)<>"#" THEN  LET a=a+1
820 IF a=ad THEN  GO TO 840
825 LET i=i+1: GO TO 815
840 LET h$=h$( TO 2*i-2)+i$+"##"+h$(2*i-1 TO ): LET z=z+LEN i$/2+1: LET
zz=1: GO SUB 1100: RETURN
900 IF LEN i$<2 THEN  RETURN
905 GO SUB 8000
910 LET ad=ad+start: LET add=VAL i$: LET a=add-ad-2
912 IF ad>adr OR add>adr THEN  RETURN
915 IF a<-128 OR a>127 THEN  RETURN
920 IF a<0 THEN  LET a=a+256
925 LET x=INT (a/16): LET y=a-16*x
930 LET ad=ad+1: LET i$="v"+STR$ ad+","+CHR$ (x+48+39*(x>9))+CHR$ (y+48
+39*(y>9))
935 GO SUB 9000: RETURN
1100 IF zz=1 THEN  GO SUB 5305: GO TO 1105
1102 GO SUB 5300
1105 LET kl=0: LET adr=start: LET pos=0
1110 LET a=0
1112 LET b=a: IF a>=LEN h$/2 THEN  GO SUB 300: RETURN
1113 LET a=a+1: IF h$(2*a)="#" THEN  GO TO 1117
1114 IF pos>=16 THEN  GO SUB 5300
1115 LET kl=kl+1: GO TO 1113
1117 PRINT AT pos+5,1;adr;" ";: LET x$=h$(2*b+1 TO 2*a)
1120 FOR q=1 TO LEN x$/2: PRINT x$(2*q-1 TO 2*q)+" ";: NEXT q
1125 LET adr=adr+kl: LET pos=pos+1: LET kl=0
1130 GO TO 1112
1200 INPUT "Name? ";v$: IF LEN v$>10 THEN  GO TO 1200
1205 INPUT "Speichern von-bis?                    ";aa,bb: IF bb<=aa THEN  GO
TO 1205
1210 SAVE v$CODE aa,bb-aa+1
5025 PRINT AT 1,0;"Programmbereich:    ";Ramt+1;"-";64900;AT 2,0;"Startad
resse:     ";start: RETURN
5100 GO SUB 5305
5110 PRINT AT pos+5,1;Adr-g+2;" ";x$: LET pos=pos+1
5120 RETURN
5300 PRINT AT 21,9;"Taste druecken": PAUSE 0
5305 PRINT AT 21,9;"                    "
5310 FOR l=5 TO 21: PRINT AT l,0;"                                  ": NEXT
 l
5315 LET pos=0
5320 RETURN
8000 LET x$=i$(2 TO ): LET sr=1
8010 LET sr=sr+1: IF x$(sr)="," THEN  GO TO 8020
8015 IF sr>LEN x$ THEN  RETURN
8017 GO TO 8010
8020 LET i$=x$(sr+1 TO ): LET ad=VAL x$( TO sr-1)-start
8025 RETURN
9000 GO SUB v(CODE i$(1)-102)
9010 GO TO 40
9999 SAVE "Quassi" LINE 1: GO TO 9999
```

* * * * * * * * * * * * * * *

# SOFTWARE-TEST CPC 464: 1. FOREST AT WORLD's END

Der CPC-464 ist wie geschaffen für gute Graphik-Adventure, er hat einen großen Umfang an Memory-Platz und gute Graphik-Eigenschaften. Die Graphik allein macht aber noch kein gutes Spiel, richtige Adventure-Spieler benötigen einen ausführlichen Text, der ihre Phantasie anregt.
In FOREST AT WORLD's END hat man alles, die Graphik sorgt für die richtige Atmosphäre, der Text für die Spannung. Und man wird schon beim Einladen auf die Folter gespannt, wie lange es denn wohl dauert. Gute Graphik benötigt einen umfangreichen Speicherplatz.
Nicht alle Orte im Adventure haben ein Bild, aber dort, wo es erscheint, ist es großartig. Die Bilder wurden von Terry Greer entworfen, ein Meister in der Computer-Graphik. Man darf sich diesen Namen merken, er hat wirklich alle Möglichkeiten des CPC 464 ausgenutzt. Die Bilder werden vom Computer gespeichert und bei Bedarf schnell auf den Bildschirm gezeichnet.
Die meisten modernen Adventure nehmen auch lange Sätze an. Die Routine, mit der das geschieht, wird PARSER genannt. Wie gut der PARSER ist, bestimmt die Spielqualität und den Spielwitz. FOREST AT WORLD's END hat noch keinen ausgereiften PARSER und ist daher in seinem Wortschatz beschränkt. Das führt manchmal zu langer Suche nach dem passenden Wort, was bei vielen Adventure-Spielern zum Reiz des Spiels gehört.
Das Kommando „to get into the boat" ist „climb into boat". „Enter boat", „go boat", „jump into boat" usw. werden nicht angenommen. Hier verbraucht man viel Zeit, um herauszufinden, welche Kommandos das Programm annimmt. Ein großer Wortschatz benötigt einen großen Platz an MEMORY und um bei der Wahrheit zu bleiben, FOREST AT WORLD's END hat eine Vielzahl gespeichert.

In Fachgeschäften oder im Versandhandel für 24,95 DM erhältlich.

# CPC 464: 2. GHOULS

Dieses Spiel befaßt sich, wie der Name sagt, mit Geistern, Gespenstern und Dämonen. Im Stil ist es vergleichbar mit dem bekannten MANIC MINER und mit ROLAND IN TIME. Und da es eine gute Graphik hat, ist es sehr vergnüglich zu spielen und dabei mit ausreichenden Schwierigkeitsgraden versehen.

Als PAC-MAN-Spieler müssen Sie viel laufen und auf hohe Plattformen springen oder auf niedrigere zurückfallen. Damit sind noch nicht alle Schwierigkeiten genannt. Denn böse Geister wollen verhindern, daß Sie das Haus erreichen. Sie starten irgendwann über Ihnen und versuchen, sich unbemerkt anzuschließen. Verhindern Sie das, sonst ist Ihr Leben verwirkt.

Immerhin beginnen Sie mit fünf Leben!

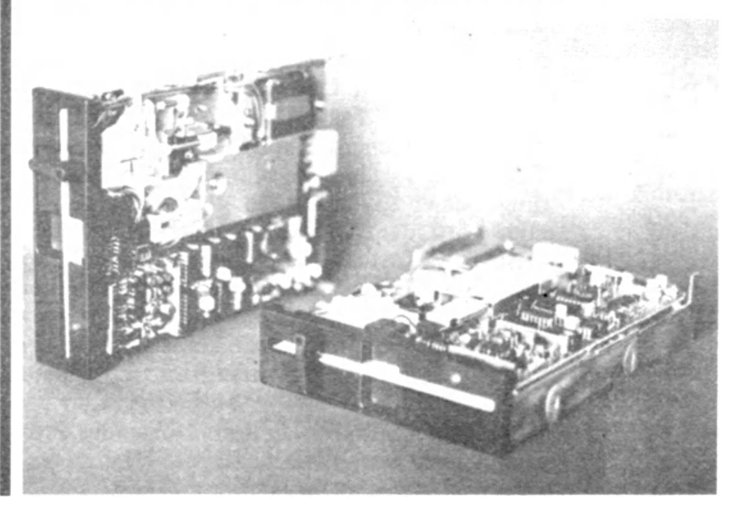
In halber Höhe des Bildschirms ist ein goldener Ring. Wenn Sie diesen erreichen, bevor der Geist Sie erreicht, werden alle Geister auf dem Bildschirm zerstört und man kann seine Aufgabe fortsetzen. Denn die Zeit ist auch limitiert und wenn Sie abgelaufen ist, haucht Ihre Spielfigur – ihr Leben aus.

Sollte es Ihnen nach einiger Übung gelingen, alle Gefahren zu meistern und das Spielende gesund und munter – bis auf den überstandenen Spielstreß – zu erreichen, so können Sie sich in eine HI-SCORE-Tabelle eintragen und damit möglicherweise die Familienehre retten.

Dies ist ein sehr gutes Spiel und es kann Sie für Stunden gefangen halten. Für DM 24,95 in guten Fachgeschäften erhältlich. Hersteller: MICROPOWER.

# SCSE vertreibt Panasonic-Laufwerkspalette

Die 3,5 Zoll- und 5 ¼ Zoll-Laufwerks-Familie von Panasonic wird von der SCSE GmbH in Aschaffenburg vertrieben.
Die 3,5 Zoll-Laufwerke sind in drei Ausführungen mit 250 kB, 500 kB und einem Megabyte lieferbar.
Bei den 5 ¼ Zoll-Laufwerken stehen zwei Versionen zur Verfügung: Ein Laufwerk ist mit 500 kB bei 48 tpi lieferbar, das andere Laufwerk ist wahlweise für 1,0 oder 1,6 Megabyte lieferbar.



# Dokumentation von Systemvariablen

## Für den ZX Spectrum

Die Systemvariablen sind die wichtigsten Speicherzellen im ZX Spectrum überhaupt. Sie enthalten die Informationen, die das System zum Verwalten aller Funktionen, Speicherteile und anderer Anwendungsmöglichkeiten braucht. Wenn der Programmierer sie vollständig beherrscht und die Möglichkeiten kennt, kann er seine BASIC-Programme um einiges verbessern und vervollkommnen. Aber sie können nicht nur dem BASIC-Programmierer, sondern gerade auch dem Maschinensprache-Programmierer nützen, da dieser immer wieder die ROM-Routinen verwendet und die Systemvariablen sozusagen die Brücke dahin sind.

Zunächst einige Grundkenntnisse, die nötig sind, um zu verstehen, was die „Systemvariablen" eigentlich sind und wie man damit arbeitet:
Der Speicher des ZX Spectrum besteht aus 65536 Speicherzellen, die von 0 bis 65535 durchnumeriert sind. Diese Nummern nennt man Adressen. Jede Adresse kann ein Byte, d. h. eine ganze Zahl von 0 bis 255 speichern. Man unterscheidet im allgemeinen zwei Speichertypen:
Das ROM (read-only-memory = Nur-Lese-Speicher) enthält feste Werte, die nicht geändert werden können. Also können aus dem ROM nur Werte gelesen werden. Beim ZX Spectrum reicht das ROM von der Adresse 0 bis 16383. In diesem Speicher befindet sich ein Programm, das die BASIC-Befehle in die sogenannte Maschinensprache umwandelt, die der Computer versteht. Dieses Programm nennt man „Interpreter".
Das RAM (random access memory = Frei-Benutzbarer-Speicher) wird beim ZX Spectrum zu vielen verschiedenen Dingen verwendet, beginnt bei der Adresse 16384 und endet bei der Adresse 65535. (Beim 16K Spectrum endet er bei 32767.) Aus diesen Speicherzellen können ebenfalls Bytes gelesen werden, aber es können auch neue Zahlen abgespeichert werden.
Im BASIC gibt es zwei Befehle, die mit Speicherzellen arbeiten: POKE Adresse, Byte schreibt einen Wert in die mit der Adresse angegebenen Speicherstelle. PRINT PEEK Adresse ergibt den Inhalt der adressierten Speicherzelle. Oft müssen größere Werte als 255 abgespeichert werden, deswegen werden dann die Werte in zwei Bytes aufgeteilt, die dann in zwei Adressen gespeichert werden. Dies geschieht mit den Befehlen POKE a,w-256*INT (w/256): POKE a+1,INT (w/256). Die Werte (w) können mit PRINT PEEK a+256*PEEK(a+1) wieder ausgelesen werden. Auf diese Weise kann man Zahlen von 0 bis 65535 abspeichern.
Als „Systemvariablen" bezeichnet man den Speicherbereich, den der Interpreter beim Arbeiten mit Adressen und Zahlen benutzt. Die Systemvariablen stehen im RAM von Adresse 23552 bis 23733, können also vom Programmierer verändert werden. Dadurch ergeben sich vielfältige und verschiedenste Möglichkeiten, seine BASIC-Programme zu erweitern und ihnen den „letzten Schliff" zu geben. Die Systemvariablen enthalten die Informationen, die der Computer zum Verwalten aller seiner Funktionen, Speicherteile und anderer Anwendungsmöglichkeiten braucht. Im folgenden also eine vollständige und ausführliche Liste der Systemvariablen.

## Die „Systemvariablen"

**Adressen** — **Bedeutung**

23552-23559: Speicherplätze, die der Interpreter beim Lesen der Tastatur benutzt. Sie sind in zwei Blöcke aufgeteilt, da manchmal zwei Tasten zugleich gedeutet werden müssen, wenn zum Beispiel Großbuchstaben eingegeben werden. Das erste Byte gibt an, ob eine Taste gedrückt wurde. Das zweite Byte ist ein Wiederholungszähler. Das dritte Byte gibt die Wartezeit vor Beginn der Wiederholung an. Das vierte Byte ist der Kode der gedrückten Taste. Die nächsten vier Bytes haben entsprechende Aufgaben für eine zweite Taste. Die folgende Systemvariable ist allerdings besser zu verwenden.

23560: Hier wird der Kode der zuletzt gedrückten Taste bzw. Tastenkombination gespeichert. So kann man also feststellen, ob während einer langen Rechenroutine eine Taste gedrückt wurde, obwohl bei der Abfrage keine Eingabe erfolgt. Dazu muß man in diese Adresse vorher 0 poken.
Übrigens kann man sich durch folgendes Programm eine komplizierte Repeat-Routine ersparen:

```
10 POKE 23560,0
20 IF PEEK 23560 THEN PRINT CHR$
PEEK 23560: GO TO 10
30 GO TO 20
```

Diese Routine hat im Gegensatz zu

```
10 PRINT INKEY$:
20 GO TO 10
```

den Vorteil, daß zuerst eine Weile gewartet wird und dann die Taste nicht zu schnell wiederholt wird. (Dies ist übrigens auch in Maschinensprache möglich.) Mit den beiden nächsten Systemvariablen können diese Wartezeiten verändert werden.

23561: Hier steht die Zeit, wie lang eine Taste gedrückt werden muß, damit sie wiederholt wird, diese beträgt normalerweise 35. (Angabe in 50stel Sekunden.) Die Zahl 0 entspricht hier 256/50 Sekunden.

23562: Hier steht die Zeit, die zwischen den Wiederholungen der Taste gewartet wird, diese beträgt normalerweise 5. (Angabe in 50stel Sekunden.) Die Zahl 0 entspricht auch hier 256/50 Sekunden.

23563/23564: Adresse bei der Bearbeitung von selbstdefinierten Funktionen.

23565-23567: Speichert unwichtige Farbinformationen.

23568-23605: Hier werden Informationen von OPEN # und CLOSE # gespeichert. Diese Befehle sind übrigens im Gegensatz zum Handbuch auch ohne Microdrive zu benutzen. Normalerweise geht jede PRINT- oder LIST-Anweisung auf den Bildschirm. Gibt man allerdings Print #0 oder #1 ein, wird der Text auf die 23ste Zeile gedruckt, wo normalerweise die Eingabe erfolgt. (Darüber mehr bei Adresse 23659.) PRINT #2 ist dasselbe wie PRINT und PRINT #3 dasselbe wie LPRINT. Durch OPEN # kann man die Funktionen von #0 bis #15 ändern. Hat man zum Beispiel ein Programm für den Drucker geschrieben und will es zunächst auf dem Bildschirm ausprobieren, braucht man nur OPEN #3, „s" einzugeben und jeder LPRINT-Befehl wird auf dem Bildschirm ausgegeben. „s" bedeutet Ausgabe auf den oberen Bildschirmteil, „k" auf den unteren und „p" auf den Drucker. CLOSE #3 stellt den Ausgangszustand wieder her.

23606/23607: Diese Speicherplätze ermöglichen einen völlig neuen Zeichensatz, denn hier steht die Adresse, an der für jeden Buchstaben und jedes Zeichen 8 Bytes abgelegt sind, genau wie bei den selbstdefinierten Graphikzeichen. Diese Systemvariable zeigt auf den Anfang der Tabelle. Da aber die Zeichen CHR$ 0 bis CHR$ 31 keine druckbaren Zeichen sind, fängt die Tabelle eigentlich 32*8 Bytes = 256 Byte später an. Dort sind dann die 8 Bytes für das Leerzeichen, dann folgt das Ausrufezeichen, die Anführungsstriche usw. Versuchen Sie mal POKE 23606,8. Dies verschiebt den Zeiger um 8 Byte = 1 Zeichen. Läßt man diese Variable ins RAM zeigen, kann man dort seinen selbst entworfenen Zeichensatz ablegen. 23606 enthält normalerweise 0 und 23607 die Zahl 60.

23608: Länge des Warntons, wie er zum Beispiel beim Drücken einer Taste nach CLEAR 23850 ertönt. Normalerweise 64.

23609: Länge des Tastaturpieps. Normalerweise 0.

23610: Hier werden Fehlernummern gespeichert. 255 = OK; 0 = NEXT without FOR; 1 = Variable not found usw. Versuchen Sie POKE 23610,0. Dies kann man aber leider im Programm nur in der letzten Zeile verwenden. Allerdings funktioniert es dafür in Maschinensprache um so besser: Die Systemvariable wird mit der gewünschten Fehlernummer geladen und dann einfach RST 8 ausgeführt.

23611-23616: Zahlen und Adressen, mit denen der Interpreter arbeitet.

23617: Setzt den Modus des Cursers: 0 = L-Mode; 1 = E-Mode; 2 = G-Mode. (C-Mode siehe Adresse 23658.) Poked man höhere Zahlen in diese Adresse, erscheinen einige recht merkwürdige Curser-Arten, die sich effektvoll in INPUTs verwenden lassen.

23618-23620: Diese drei Adressen ermöglichen einen Sprung zu einem beliebigen Befehl in einer Zeile und nicht nur zum ersten, wie bei GO TO. Auch die Befehle nach THEN können aufgerufen werden. Um solche Befehle anzusprechen, poked man zuerst die Zeilennummer in die Adressen 23618/23619 und dann in die Adresse 23620 die Stelle des Befehls in der Zeile. Der erste Befehl in der Zeile hat die Nummer 1, der nächste Befehl nach einem „:" oder einem „THEN" die Nummer 2 usw. Existiert kein Befehl an der angegebenen Stelle, wird die Fehlermeldung „statement lost" gedruckt.

23621-23623: Zeilennummer und Befehlsstelle des gerade bearbeiteten Befehls.

23624: Diese Variable wird von jedem BORDER-Befehl gesetzt, aber kann auch benutzt werden, um die Farben des unteren Bildschirmteils zu setzen. Die Zahl berechnet sich aus 128*FLASH + 64 BRIGHT + 8*PAPERfarbe + INKfarbe. Damit sind auch verschiedenfarbige INPUTs möglich. Nach jeder Eingabe hat der Rand die gleiche Farbe, wie die hier gespeicherte PAPER-Farbe richtet sich aber nicht nach FLASH und BRIGHT.

23625/23626: Hier steht die Zeilennummer, die bei EDIT editiert wird.

23627/23628: Hier steht die Anfangsadresse des Variablenspeichers. Systemvariablen, die den Speicheraufbau festlegen, sollten vom Programmierer nicht verändert werden, da der Interpreter dann nicht mehr richtig arbeiten kann.

23629/23630: Adresse einer gerade verwendeten Variable.

23631/23632: Anfangsadresse der Kanalinformationen.

23633/23634: Adresse einer verwendeten Kanalinformation.

23635/23636: Anfangsadresse des Basicspeichers, welcher folgendermaßen aufgebaut ist: Für jede Zeile zunächst zwei Bytes, die die Zeilennummer angeben. Achtung: Diese beiden Bytes stehen ausnahmsweise in vertauschter Reihenfolge! (Zuerst das höherwertige und dann das niederwertige.) Dann folgen zwei Bytes, die die Länge des Zeilentextes angeben. (Wieder in der üblichen Reihenfolge.) Danach stehen die Kodes der Befehle und Zeichen und am Schluß der Zeile steht 13 (= ENTER). Übrigens nach jeder dezimalen Zahl im Text steht 14, gefolgt von 5 Bytes, die diese Zahl in ihrer binären Form angeben, da der Computer nur damit arbeiten kann. (Diese 6 Bytes werden beim normalen LISTing verschluckt.)

23637/23638: Gibt die Adresse der Basiczeile an, die der gerade bearbeiteten folgt.

23639/23640: Diese Adresse zeigt auf das erste Byte, nach der zuletzt gelesenen „DATA-Zahl". In dieser Adresse steht dann also entweder 44 (= Trennungskomma) oder 13 (= Ende der Zeile). Wenn noch kein READ verwendet wurde, ergibt diese Systemvariable die letzte Adresse vor dem Basicspeicher, ist also um eins kleiner als die in 23635/23636 abgespeicherte Zahl.

23641/23642: Ab der hier angegebenen Adresse steht der Text der Zeile, die gerade eingegeben wird.

23643/23644: Adresse des Cursers im vorherigen Speicherbereich. Bei einem INPUT steht in dieser Adresse, an welcher Stelle im eingegebenen Text der Curser zuletzt stand. (Am Anfang = 1 usw.)

23645/23646: Adresse des Zeichens, das beim Ausführen einer Zeile als nächstes bearbeitet wird.

23647/23648: Adresse des Zeichens nach einem „?"-Curser.

23649/23650: Anfangsadresse des INPUT-Speichers.

23651/23652: Anfangsadresse des Arbeitsspeichers des ROM-Rechenteils (Kalkulator).

23653/23654: Anfangsadresse des Speichers, der zur Zeit keine besondere Aufgabe hat, also vom Programmierer frei verfügbar ist. (Dazu auch 23730/23731.)

23655: Speicherstelle des Kalkulators (des ROM-Rechners).

23656/23657: Adresse, an der der Kalkulator bis zu 6 Fliesskommazahlen zwischenspeichert. Normalerweise 23698.

23658: Diese Variable ermöglicht das Umschalten auf den „C"-Curser (POKE 23658,8). Um den normalen Curser wieder zu erhalten, muß man diese Adresse auf 0 setzen.

23659: Angabe der Zeilenzahl des unteren Bildschirmteils. Diese Speicherzelle wird von jeder Fehlmeldung und jedem INPUT auf 2 gesetzt (Zeile 22 ist leer und Zeile 23 wird bedruckt). Man kann die 22ste Zeile ganz normal mit PRINT AT 22,0; benutzen, wenn in diese Adresse 1 gepoked wird. Man kann aber auch den INPUT auf eine andere Zeile setzen, indem man INPUT AT n,0; AT y,x;a$ eingibt. n+1 ist die Anzahl der unteren Zeilen, y gibt die Zeile des unteren Bildschirmteils an, dessen Zeilen hier von 0 bis n durchnumeriert sind. x gibt schließlich noch die Spalte an. Enthält diese Speicherzelle 0 oder eine Zahl, die größer ist als 24, darf nicht auf den unteren Bildschirmteil zugegriffen werden, da sonst der Interpreter nicht weiterarbeitet und keine Taste mehr annimmt oder einfach NEW ausgeführt. (Dies ist also gut als BREAK-Schutz in einem Basicprogramm zu verwenden, da die BREAK-Meldung ja auch dorthin gedruckt wird.) Nun kann man aber auch auf den unteren Bildschirm drucken: PRINT #1; AT y,x; Die Zeile errechnet sich wieder von der obersten Zeile des unteren Bildschirms, die man ja mit dieser Systemvariable verändern kann. Wurde der untere Bildschirm noch nicht verwendet, wird PRINT #1;a$ automatisch unten AT 1,0; gedruckt. Ansonsten kann man PRINT #1 genauso wie PRINT verwenden. INPUT "" löscht den unteren Bildschirm und setzt diese Variable wieder auf 2.

23660/23661: Gibt die Zeilennummer der obersten Zeile beim automatischen Listing an (nach der Eingabe von ENTER ohne LIST).

23662/23663: Zeilennummer in einem Basicprogramm, zu der CONTINUE nach einem BREAK springt.

23664: Befehlstelle in der Zeile, die in der vorherigen Systemvariable angegeben wurde, zu dem CONTINUE nach einem BREAK springt.

23665: Zwischenspeicher des Interpreters.

23666/23667: Wird bei der Bearbeitung von Strings benutzt.

23668/23669: Zeiger für eine Tabelle im ROM, mit der der Interpreter die Syntaxprüfung durchführt.

23670/23671: Diese Speicherzellen werden von der Zahl hinter RANDOMIZE gesetzt. Bei RANDOMIZE oder RANDOMIZE 0 werden die Inhalte der Speicherzellen 23672/23673 genommen.

23672–23674: Diese Speicherzellen enthalten beim Einschalten des Gerätes alle 0. 23672 wird alle 50stel Sekunden um eins erhöht. Wenn diese Adresse 255 erreicht, wird sie auf 0 gesetzt und 23673 um eins erhöht. Das gleiche gilt bei 23673/23674. Die Einschaltzeit berechnet sich also mit PRINT INT (( PEEK 23672+256* PEEK 23673+65535* PEEK 23674)/50). Dieser Zählvorgang wird allerdings von jedem BEEP und den Cassettenroutinen unterbrochen.

23675/23676: Enthält die Anfangsadresse der selbstdefinierbaren Graphikzeichen. Wenn man zum Beispiel seinen 16K Spectrum auf 48K erweitert und Graphikzeichen vorher mit CODE auf Cassette abgespeichert hat, braucht nur POKE 23676,127 eingegeben zu werden und man hat die Graphikzeichen wieder zur Verfügung. Außerdem kann man hiermit im Programm zwischen mehreren selbstdefinierten Graphikzeichensätzen umschalten und ist nicht mehr auf die „wenigen" 21 beschränkt.

23677: X-Koordinate des zuletzt gePLOTeten Punktes.

23678: Y-Koordinate des zuletzt gePLOTeten Punktes.

23679–23683: Hier werden Positionen bei der Ausgabe gespeichert.

23684/23685: PRINT-Position im Bildschirmspeicher.

23686/23687: PRINT-Position des unteren Bildschirmteils im Bildschirmspeicher.

23688–23691: Wird bei der Berechnung von PRINT-Positionen verwendet.

23692: Diese Speicherzelle gibt an, wie oft der Bildschirm um eine Reihe nach oben geschoben werden soll, bevor nach „scroll?" gefragt wird. 0 heißt hierbei 256mal. Übrigens kann man einen schnellen Bildschirmscroll mit RANDOMIZE USR 3582 erreichen, ohne etwas zu drucken. Da RANDOMIZE sich aber auf Zufallszahlen auswirkt, ist es manchmal zu empfehlen zum Beispiel LET u=USR 3582 oder IF USR 3582 THEN zu verwenden.

23693: Hier werden die Farben gespeichert, die mit den Farbbefehlen gesetzt wurden: 128*FLASH +64*BRIGHT +8*PAPERfarbe +INKfarbe. Mit einem POKE kann man sich also mehrere Befehle ersparen.

23694: Außergewöhnliche Farbinformationen:
128, wenn FLASH 8
\+ 64, wenn BRIGHT 8
\+ 56, wenn PAPER 8
\+ 7, wenn INK 8

23695: Enthält bei der Ausführung von PRINT zunächst das gleiche wie 23693, und wird dann nach den Farbbefehlen im PRINT geändert.

23696: Enthält bei der Ausführung von PRINT zunächst das gleiche wie 23694, und wird dann nach den Farbbefehlen im PRINT geändert.

23697: Weitere Farbinformationen:
192, wenn PAPER 9
\+ 48, wenn INK 9
\+ 12, wenn INVERSE 1
\+ 3, wenn OVER 1
Für Maschinencode-Programmierer: BIT

7=PAPER 9, BIT 5=INK 9, BIT 3=INVERSE 1, BIT 1=OVER 1. Die anderen Bits können im PRINT verändert werden.

23698–23727: Normalerweise benutzt der Kalkulator diese Speicherzellen, um bis zu sechs Fließkommazahlen zwischenzuspeichern, die je 5 Bytes verbrauchen. (Dazu auch 23656/23657.)

23728/23729: Wenn ein externes Gerät an den ZX Spectrum angeschlossen ist und ein Unterbrechungssignal sendet, wird NEW ausgeführt. Wenn in diesen Speicherzellen andere Werte als 0 stehen, passiert nichts.

23730/23731: Adresse nach dem letzten Byte des frei verfügbaren Speichers. Der freie Speicherplatz berechnet sich übrigens aus PRINT 65535-USR 7962. Mit dieser Formel erhält man den am besten gesicherten Wert, da es dasselbe Verfahren ist, das auch der Interpreter verwendet.

23732/23733: Adresse des letzten Bytes überhaupt im RAM. Beim 16K Spectrum als 32767 und beim 48K Spectrum 65535.

# Mailbox-Telefonbuch

Bei Warnekes Mail-Telefonbuch handelt es sich um einen neuen Zubehörartikel der Computerbranche. Zur Zeit enthält dieses DFÜ-Anbieter-Verzeichnis ca. 150 verschiedene Mail-Box-Anschlüsse. Für 1986 ist eine wesentlich erweiterte Ausgabe geplant.

Zur Benutzung des Buches zwei Hinweise: In den Rubriken „Betriebszeit" und „Bemerkungen" erscheinen teilweise die Begriffe „wird getestet" und „wird geprüft". Bei Redaktionsschluß lagen hier noch keine exakten Informationen vor, deshalb erscheinen in diesen Fällen die o. a. Begriffe.

Für die Rubrik „Bemerkungen" ist geplant, z. B. den Eröffnungsmodus der Mailbox und die Paßwörter etc. sowie eine kleine Beschreibung des Betriebssystems und des Operators einzutragen.

CPC-464

# Befehlserweiterung des CPC-Befehlssatzes um einen Box-Befehl

Dieses Assembler und Basicprogramm bewirkt eine Befehlserweiterung des CPC-Befehlssatzes um einen BOX-Befehl.

Nach einmaligem Aufruf der Maschinencoderoutine kann diese gelöscht werden und der Befehl befindet sich dann in der RSX-Erweiterung im RAM.

Die Parameter des Befehls lauten wie folgt:

x1 + y1 – Startposition befindet sich Ecke links unten

x2 + y2 – Endposition befindet sich Ecke rechts oben.

Der letzte Parameter-Befehl bestimmt die Farbe.

```
10 '*********************************
20 '***** !BOX BEFEHL FUER CPC ******
30 '*****  M.BANGEMANN  1985   ******
40 '*********************************
50 '
60 'BASIC LADER FUER MC PROGRAMM
70 '
80  h=HIMEM
90  MEMORY &806F
100 FOR i=&8070 TO &80CB
110 READ x:POKE i,x
120 NEXT
130 DATA &01,&79,&80,&21,&82,&80,&c3,&d1,&bc,&7e,&80,&c3,&86,&80,&42,&4f
140 DATA &d8,&00,&00,&00,&00,&00,&fe,&05,&c0,&cd,&c6,&bb,&d5,&e5,&dd,&7e,&00
150 DATA &cd,&de,&bb,&dd,&56,&09,&dd,&5e,&08,&dd,&66,&07,&dd,&6e,&06,&e5,&d5
160 DATA &d5,&cd,&c0,&bb,&d1,&d5,&dd,&66,&03,&dd,&6e,&02,&e5,&cd,&f6,&bb,&e1
170 DATA &dd,&56,&05,&dd,&5e,&04,&d5,&cd,&f6,&bb,&d1,&e1,&cd,&f6,&bb,&d1,&e1
180 DATA &cd,&f6,&bb,&e1,&d1,&c3,&c0,&bb
190 CALL &8070
200 'Sichern der Befehlserweiterung auf Cassette oder Disc
210 MODE 2:INK 0,13:BORDER 13:INK 1,0
220 LOCATE 1,12:PRINT"(D)isc oder (C)assette ?"
230 a$=INKEY$:IF a$="" THEN 230
```

```
240 IF a$="D" OR a$="d" THEN PRINT"Bitte formatierte Diskette einlegen und Taste
 druecken" ELSE 253
250 CALL &BB18:GOTO 256
253 ON ERROR GOTO 256
255 |TAPE
256 SAVE"BOX.BIN",B,&8070,123
270 INK 1,1,0:MODE 2:LOCATE 1,12
280 PRINT"  Vor weiteren Eingaben MEMORY ";h;" eingeben !!!"
290 NEW
```

## Demonstrationsprogramm

```
10 MODE 1
20 INPUT "Schrittweite";sc
30 MODE 0
40 x1=0:y1=0:x2=639:y2=399
50 FOR durch=1 TO 320 STEP sc
60 |BOX,x1,y1,x2,y2,durch/sc
70 x1=x1+sc:y1=y1+sc:x2=640-x1:y2=400-y1
80 NEXT
90 CALL &BB18
100 CLEAR:GOTO 10
```

```
                     10 ;**************************************
                     20 ;****  RSX - ERWEITERUNG  DES CPC ******
                     30 ;****  NACHLADBARER  BOX - BEFEHL ******
                     40 ;****       COPYRIGHT  1985       ******
                     50 ;****  MARTIN  BANGEMANN   NAGOLD ******
                     60 ;**************************************
                     70 ;
                     80 ;
BBDE                 90 SETPEN:  EQU   #BBDE               ;Penfarbe setzen
BBF6                100 DRAWAB:  EQU   #BBF6               ;Linie ziehen
BCD1                110 LOGEXT:  EQU   #BCD1               ;Erweiterung einbinden
BBC0                120 MOVABS:  EQU   #BBC0               ;Grafikcursor setzen
BBC6                130 ASKCUR:  EQU   #BBC6               ;Ausgangsposition Cursor
                    140
8070                150          ORG   #8070               ;MC-Startadresse
8070  017980        160          LD    BC,RSX              ;Box in die Erweiterung e
intragen
8073  218280        170          LD    HL,KERNEL
8076  C3D1BC        180          JP    LOGEXT
8079  7E80          190 RSX:     DEFW  TABLE
807B  C38680        200          JP    BOX
807E  424F          210 TABLE:   DEFM  "BO"
8080  D8            220          DEFB  "X"+#80
8081  00            230          DEFB  0
8082                240 KERNEL:  DEFS  4
8086  FE05          250 BOX:     CP    5                   ;5 Parameter holen
8088  C0            260          RET   NZ
8089  CDC6BB        270          CALL  ASKCUR
808C  D5            280          PUSH  DE
808D  E5            290          PUSH  HL                  ;Ausgangsposition merken
808E  DD7E00        300          LD    A,(IX)
8091  CDDEBB        310          CALL  SETPEN              ;Penfarbe setzen
8094  DD5609        320          LD    D,(IX+9)            ;linke untere Ecke (x1)
8097  DD5E08        330          LD    E,(IX+8)
809A  DD6607        340          LD    H,(IX+7)            ;linke untere Ecke (y1)
809D  DD6E06        350          LD    L,(IX+6)
80A0  E5            360          PUSH  HL                  ;x1 y1 merken
80A1  D5            370          PUSH  DE
80A2  D5            380          PUSH  DE
80A3  CDC0BB        390          CALL  MOVABS
80A6  D1            400          POP   DE
80A7  D5            410          PUSH  DE
80A8  DD6603        420          LD    H,(IX+3)            ;y2 von rechter oberer Ec
ke
```

```
80AB  DD6E02     430        LD    L,(IX+2)
80AE  E5         440        PUSH HL               ;y2 merken
80AF  CDF6BB     450        CALL DRAWAB           ;Linie von (xly1) nach (x
ly2)
80B2  E1         460        POP  HL               ;y2 holen
80B3  DD5605     470        LD    D,(IX+5)        ;x2 rechte obere Ecke
80B6  DD5E04     480        LD    E,(IX+4)
80B9  D5         490        PUSH DE               ;x2 merken
80BA  CDF6BB     500        CALL DRAWAB           ;Linie von (xly2) nach (x
2y2)
80BD  D1         510        POP  DE               ;x2 holen
80BE  E1         520        POP  HL               ;yl holen
80BF  CDF6BB     530        CALL DRAWAB           ;Linie von (x2y2) nach (x
2yl)
80C2  D1         540        POP  DE               ;xl holen
80C3  E1         550        POP  HL               ;yl holen
80C4  CDF6BB     560        CALL DRAWAB           ;Linie zum Ursprung

Hisoft GENA3.1 Assembler. Page      2.

80C7  E1         570        POP  HL               ;Cursorausgangsposition
80C8  D1         580        POP  DE               ;wiederherstellen
80C9  C3C0BB     590        JP   MOVABS

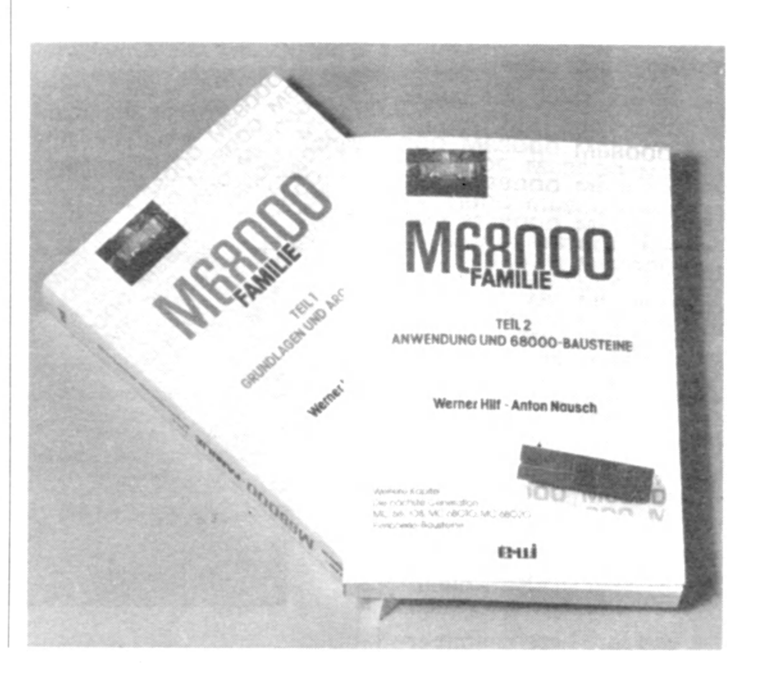
Pass 2 errors: 00

Table used:    123  from   282
```

# M 68000-Familie

## Teil 1: Grundlagen und Architektur – Teil 2: Anwendung und 68000-Bausteine

Dieses zweibändige Werk befaßt sich grundlegend mit der M 68000-Familie, einer Familie von leistungsfähigen 16/32-Bit-Prozessoren. Der Leser findet alle notwendigen Informationen über den M 68000. Angefangen von der Entstehungsgeschichte über die detaillierte Hardwarebeschreibung bis zu den Adressierungsarten und dem ausführlichen Befehlssatz. Es eignet sich sowohl als Lehrbuch als auch als Nachschlagwerk.
In Band 1 werden die Grundlagen geschaffen und die Architektur (Hard- und Software) mehr oder weniger theoretisch behandelt. Entstehungsgeschichte und Zweithersteller finden Sie in Kapitel 1. Grundlagen über eine microcodierte CPU und die Beschreibung der Pins des 68000-Bausteines sind in Kapitel 2 ausführlich dargestellt. Wichtige Ausdrücke und Definitionen, die Ihnen in beiden Bänden begegnen werden, sind gleichfalls in Kapitel 2 niedergeschrieben. Adressierungsarten und Befehlssatz folgen mit sehr vielen Beispielen in Kapitel 3 bzw. 4. Das letzte Kapitel dieses Bandes setzt sich mit den 68000-Versionen auseinander – den sogenannten Masken.
Der Band eignet sich unter anderem für diejenigen, die sich schon eingehender mit M 68000 beschäftigt haben.
In Band 2 findet der Leser zusätzliche Informationen und praktische Hinweise für das Arbeiten mit dem M 68000. So kommen die Software (Kapitel 6: Assemblerprogrammiertechnik mit M 68000-Befehlen) und die Hardware (Kapitel 9: Ein M 68000-Single-Bordsystem) auf keinen Fall zu kurz. Die Kapitel 7 und 8 beschäftigen sich mit Peripherie-Bausteinen und mit neuen bzw. zukünftigen Mitgliedern der 68000-Familie. In Kapitel 10 wird der VME-Bus beschrieben.
Beide Bände zusammen vermitteln über den eigentlichen Prozessor hinaus Informationen, die von Grundlagen neuer Programmiertechniken oder Betriebssysteme über Peripherie-Bausteine bis zur nächsten Generation der 68000-Produkte reichen.



# ALIEN ATTACK

## Ein schnelles Weltraumspiel für MSX-Computer

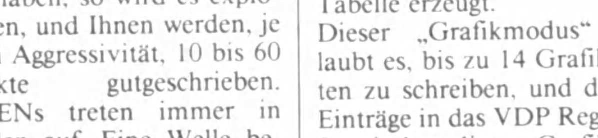
Geben Sie als erstes die Zeilen 340 und 350 sowie die Zeilen 2000 bis 2060 ein. Diese Programmzeilen ermöglichen einen ordnungsgemäßen Abbruch des Programmes sowie die korrekte Ausgabe von Fehlermeldungen.
Zur weiteren Programmeingabe verwenden Sie den Befehl AUTO 100 und die Zeilennumerierung wird automatisch richtig vorgegeben.
In Zeile 1170 finden Sie eine Menge Grafikzeichen, deren Eingabe relativ schwierig ist. Zur Eingabeerleichterung ist jedoch in Zeile 1180 die Tastenfolge beschrieben:
g bedeutet (GRAPH) Taste
c bedeutet (CODE) Taste
S bedeutet (SHIFT) Taste
Danach folgt der Buchstabe oder das Zeichen, welches ebenfalls gedrückt werden muß.
So bedeutet z. B. [gSM] daß die Tasten (GRAPH) (SHIFT) und (M) gleichzeitig gedrückt werden müssen.

Nach der Programmeingabe starten Sie das Programm mittels der Funktionstaste (F5). Sie bekommen einen Hinweis auf den Bildschirm, und müssen einen Augenblick warten, da der Rechner 4 Grafikseiten zu belegen hat. Dann erscheint das Titelbild, und die Musik ertönt.

Die Musik spielt die ganze Zeit über, kann jedoch zu jedem Zeitpunkt durch einen Druck auf die (SELECT) Taste abgestellt werden. Ein erneuter Druck auf (SELECT) stellt die Musik wieder an.
„ALIEN ATTACK" kann mit jedem beliebigen Joystick, oder auch mittels der Cursortasten gespielt werden. Um auszuwählen womit Sie spielen wollen und um das Spiel zu starten, drücken Sie einfach den entsprechenden Aktionsknopf, oder die Leertaste, wenn Sie mittels der Cursortasten spielen wollen.
Die Titelschrift verschwindet, und Ihre Laserkanone erscheint am unteren Bildschirmrand.
Kurz darauf erscheint auch das erste ALIEN, welches Sie abschießen müssen.
Vorsicht, die ALIENs verfolgen Ihre Laserkanone und versuchen diese Kanone selbst abzuschießen. Das Spiel kennt 5 verschiedene Arten von ALIENs und hat 6 Level, wobei sich die Geschwindigkeit steigert.

Steuern Sie mittels des Joysticks (oder Cursortasten) Ihre Laserkanone nach rechts oder links. Einen Schuß lösen Sie mittels des Aktionsknopfes (oder Leertaste) aus.

Wenn Sie ein ALIEN getroffen haben, so wird es explodieren, und Ihnen werden, je nach Aggressivität, 10 bis 60 Punkte gutgeschrieben.

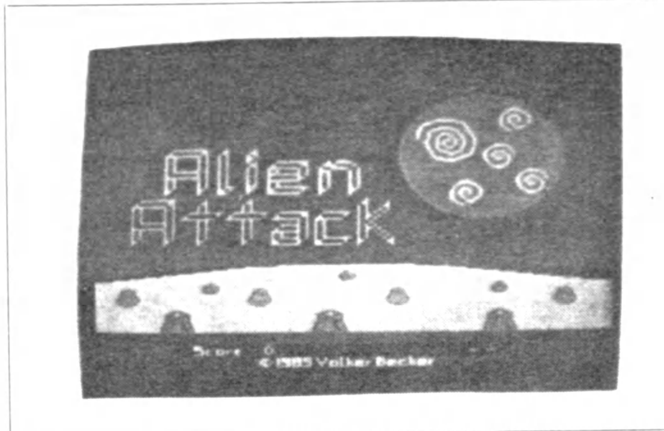


ALIENs treten immer in Wellen auf. Eine Welle besteht aus 4+2*Level ALIENs. Das heißt in Level 1 haben Sie gegen 6 ALIENs zu bestehen, in Level 6 müssen Sie es mit 16 ALIENs aufnehmen.

Wenn Sie eine Welle bestanden haben, so kommt eine andere Art von ALIENs gegen die Sie kämpfen müssen. Wenn Sie Level 6 überstanden haben, so wechseln zwar die Arten der ALIENs alle 16 Treffer, aber die Geschwindigkeit bleibt.

Die ALIENs haben die unangenehme Eigenschaft auf Ihre Laserkanone zu schießen.

Anfangs haben Sie 3 Kanonen zur Verfügung (eine im Spiel und 2 als Reserve).
Wird Ihre Kanone von einem ALIEN getroffen, so explodiert diese. Wenn alle 3 Kanonen zerstört wurden, erscheint „GAME OVER" auf dem Bildschirm.
Um ein neues Spiel zu starten, drücken Sie wieder einen beliebigen Aktionsknopf, und das Spektakel geht von vorne los.
ALIEN ATTACK arbeitet im 32 x 24 Zeichen Textmodus (SCREEN 1) und generiert die komplette Grafik durch Zeichensatz Umbelegung.
Die Farben werden mittels Eintragungen in die Farbcode-Tabelle erzeugt.
Dieser „Grafikmodus" erlaubt es, bis zu 14 Grafikseiten zu schreiben, und durch Einträge in das VDP Register 2 zwischen diesen Grafikseiten umzuschalten.
ALIEN ATTACK benutzt nun lediglich 4 dieser Grafikseiten, zwischen denen ca. alle 0.25 Sekunden umgeschaltet wird.
Alle 4 Grafikseiten sind in ihrer Grundgrafik gleich, zeigen jedoch Veränderungen.
Da eine Grafikseite einen kompletten Bildschirm umfaßt, und jetzt vier verschiedene Bilder relativ schnell hintereinander geschaltet werden, erhalten wir einen Zeichentrickfilm ähnlichen Effekt.





### Impressum

CPU
erscheint monatlich im
Tronic-Verlag, 3444 Wehretal 1

**Redaktion:**
Axel Crede (verantwortlich)
Frank Brall, Siegfried Görk,
Hartmut Wendt, Holger Grede, Ottfried Schmidt

**Freie Mitarbeiter:**
Volker Becker, Rolf Freitag

**Gesamtherstellung:**
Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

**Vertrieb:**
Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Telefon 0 61 21 / 26 60

**Anfragen nicht an Vertrieb oder Druckerei, sondern nur an den Verlag!**

**Anzeigenleitung:**
Annelie Kratzenberg, Heike Lux

**Erscheinungsweise:**
Erstverkaufstag von
CPU ist Mitte des Monats

**Anzeigenpreise:**
Bitte Mediaunterlagen anfordern

**Anzeigenannahmeschluß:**
Jeweils 3 Wochen vor Erscheinungstermin





**Bezugspreis:**
Einzelheft 5,50 DM
Abonnement: Inland 55,- DM im Jahr
(12 Ausgaben)
Ausland: Europa 80,- DM, USA 110,- DM

**Autoren, Manuskripte:**
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen.
Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,- DM pro abgedruckter Seite im Heft einverstanden sind.
Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger.
Rücksendung erfolgt nur gegen Erstattung der Kosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:
Kopierfähige Kassette oder Diskette mit dem Programm (Computer-Bezeichnung), von Drukker erstelltes Listing oder Serie von Bildschirmfotos (keine Schreibmaschinenlistings), evtl. Bildschirmfotos von einem Probelauf und ausführliche Programmbeschreibung (Erklärung der Variablenliste, Beschreibung des Bildschirmaufbaues, Farbe, Grafik usw.). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

```
100 ' ┌──────────────────────────────────┐
110 '│  A L I E N   A T T A C K  │
120 '│          ┌──────┐          │
130 '├──────────┤ M S X ├──────────┤
140 '│          └──────┘          │
150 '│ (c) 1985 by Volker Becker │
160 '│    Steinbacher Str. 10    │
170 '│    6370 Oberursel/TS.6    │
180 ' └──────────────────────────────────┘
190 '
200 DATA 29,1,3,0,29,1,3,0,254,0,87,3,254,0,87,3,29,1,166,2,29,1,166,2,29,1,58,2
,29,1,58,2,29,1,87,3,29,1,87,3,83,1,166,2,83,1,166,2,83,1,58,2,83,1,58,2,124,1,8
7,3,124,1,87,3,171,1,166,2
210 DATA 171,1,166,2,83,1,58,2,83,1,58,2,29,1,87,3,29,1,87,3,29,1,166,2,29,1,166
,2,29,1,58,2,29,1,58,2,252,1,248,3,252,1,248,3,171,1,87,3,171,1,87,3,171,1,58,2,
171,1,58,2,124,1,248,3
220 DATA 124,1,248,3,171,1,87,3,171,1,87,3,124,1,58,2,124,1,58,2,83,1,87,3,83,1,
87,3,29,1,166,2,29,1,166,2,29,1,58,2,29,1,58,2,29,1,87,3,29,1,87,3,29,1,166,2,29
,1,166,2,29,1,58,2,29,1
230 DATA 58,2,254,0,87,3,254,0,87,3,29,1,166,2,29,1,166,2,29,1,58,2,29,1,58,2,29
,1,87,3,29,1,87,3,83,1,166,2,83,1,166,2,83,1,58,2,83,1,58,2,124,1,87,3,124,1,87,
3,171,1,166,2,171,1,166,2
240 DATA 83,1,58,2,83,1,58,2,29,1,87,3,29,1,87,3,29,1,166,2,29,1,166,2,29,1,58,2
,29,1,58,2,252,1,248,3,252,1,248,3,171,1,87,3,171,1,87,3,171,1,58,2,171,1,58,2,1
24,1,248,3,124,1,248,3
250 DATA 171,1,87,3,171,1,87,3,124,1,58,2,124,1,58,2,83,1,87,3,83,1,87,3,171,1,1
66,2,171,1,166,2,124,1,58,2,124,1,58,2,171,1,87,3,171,1,87,3,171,1,166,2,171,1,1
66,2,171,1,58,2,171,1
260 DATA 58,2,252,1,248,3,252,1,248,3,171,1,87,3,171,1,87,3,171,1,58,2,171,1,58,
2,124,1,248,3,124,1,248,3,171,1,87,3,171,1,87,3,124,1,58,2,124,1,58,2,83,1,87,3,
83,1,87,3,29,1,166,2,29,1
270 DATA 166,2,29,1,58,2,29,1,58,2,29,1,87,3,29,1,87,3,29,1,166,2,29,1,166,2,29,
1,58,2,29,1,58,2,254,0,87,3,254,0,87,3,29,1,166,2,29,1,166,2,29,1,58,2,29,1,58,2
,29,1,87,3,29,1,87,3,83,1
280 DATA 166,2,83,1,166,2,83,1,58,2,83,1,58,2,124,1,87,3,124,1,87,3,171,1,166,2,
171,1,166,2,83,1,58,2,83,1,58,2,29,1,87,3,29,1,87,3,29,1,166,2,29,1,166,2,29,1,5
8,2,29,1,58,2,252,1,248,3
290 DATA 252,1,248,3,171,1,87,3,171,1,87,3,171,1,58,2,171,1,58,2,124,1,248,3,124
,1,248,3,171,1,87,3,171,1,87,3,124,1,58,2,124,1,58,2,83,1,87,3,83,1,87,3,171,1,1
66,2,171,1,166,2,124,1
300 DATA 58,2,124,1,58,2,171,1,87,3,171,1,87,3,171,1,166,2,171,1,166,2,0,0,0,0
310 CLEAR 600
320 DEFINT D,I,J,K,L,T,W
330 DEFSTR A
340 ON STOP GOSUB 2030:STOP ON
350 ON ERROR GOTO 2040
360 DIM DP(4)
370 GOSUB 1090
380 J=40:FOR I=0 TO 2:PUT SPRITE I,(J,0),8,I:J=J+32:NEXT
390 J=40:FOR I=3 TO 5:PUT SPRITE I,(J,48),7,I:J=J+32:NEXT
400 J=24:FOR I=6 TO 9:PUT SPRITE I,(J,80),3,I:J=J+32:NEXT
410 VDP(1)=227:BC=0
420 ON INTERVAL=12 GOSUB 830:RESTORE:SOUND 7,56:MS%=-1:INTERVAL ON
430 FOR JV=0 TO 2:J=RND(64):IF STRIG(JV) THEN 450
440 NEXT :GOTO 430
450 DA=0:TZ=0:BC=-1
460 FOR I=0 TO 9:PUT SPRITE I,(0,0),0,63:NEXT:ON STRIG GOSUB 1080,1080,1080
470 VDP(1)=226:BP=0:P=0:PK$=STR$(P):LG=3:TA=0:KP=120:LK=0:DN=0:DL=0:LV=1:PUT SPR
ITE 0,(KP,148),6,10:STRIG(JV) ON
480 IF BC THEN 480
490 DS=RND(200)*200+24:DA=-1:STRIG(JV) ON
500 ON STICK(JV) GOSUB 800,810,810,810,800,820,820,820
510 VPOKE &H1B01,KP
520 IF DE THEN 550
```

```
530 IF DD THEN 610
540 GOTO 500
550 LG=LG-1:TZ=0:BP=-1
560 SOUND 6,30:SOUND 7,28:PUT SPRITE 0,(KP,148),9,36:FOR I=15 TO 0 STEP -1:SOUND
 10,I:PUT SPRITE 0,(KP,148),9,39-(I)\4:FOR K=1 TO 20:NEXT K,I
570 SOUND 10,0:SOUND 7,56:DE=0
580 IF BP THEN 580
590 IF LG=0 THEN 740
600 KP=120:PUT SPRITE 0,(KP,148),6,10:GOTO 500
610 P=P+10*LV:IF P>999999! THEN P=999999!
620 PK$=STR$(P):BP=-1:TZ=0
630 SOUND 6,10:SOUND 7,28:PUT SPRITE 1,(DS,32+DV),11,32:FOR I=15 TO 0 STEP -1:SO
UND 10,I:PUT SPRITE 1,(DS,32+DV),11,35-(I)\4:FOR K=1 TO 20:NEXT K,I:PUT SPRITE 1
,(0,0),0,63
640 DD=0:SOUND 10,0:SOUND 7,56:DE=0:TA=TA+1
650 IF BP THEN 650
660 IF TA=4+2*LV THEN 670 ELSE 490
670 DN=DN+1:IF DN>4 THEN DN=0
680 LV=LV+1:IF LV>6 THEN LV=6
690 TA=0:GOTO 490
700 FOR J=1 TO LEN(PK$):VPOKE &H400*JB+713+J,ASC(MID$(PK$,J,1)):NEXT
710 TZ=TZ+1:IF TZ=4 THEN BP=0
720 VPOKE &H400*JB+729-LG,32
730 RETURN
740 STRIG(JV) OFF:DA=0:FOR I=0 TO 3:PUT SPRITE I,(0,0),0,63:NEXT:VDP(1)=227
750 SP=16:FOR I=0 TO 3:PUT SPRITE I,(SP,16),8,40+I:PUT SPRITE 4+I,(SP,50),12,I+4
4:SP=SP+32:NEXT
760 GOTO 430
770 FOR J=715 TO 720:VPOKE &H400*JB+J,32:NEXT:FOR J=727 TO 728:VPOKE &H400*JB+J,
120:NEXT:VPOKE &H400*JB+715,48
780 TZ=TZ+1:IF TZ=4 THEN BC=0
790 RETURN
800 RETURN
810 KP=ABS((KP+3)*(KP<224)+224*(KP>=224)):RETURN
820 KP=ABS((KP-3)*(KP>16)+16*(KP<=16)):RETURN
830 IF INKEY$=CHR$(24) THEN MS%=NOT MS%
840 IF MS% THEN SOUND 8,10:SOUND 9,9 ELSE SOUND 8,0:SOUND 9,0:GOTO 870
850 READ M1%,M2%,M3%,M4%:IF M1%=0 THEN RESTORE:GOTO 850
860 SOUND 0,M1%:SOUND 1,M2%:SOUND 2,M3%:SOUND 3,M4%
870 IF BP THEN GOSUB 700
880 IF BC THEN GOSUB 770
890 VDP(2)=JB:JB=JB+1:IF JB>3 THEN JB=0
900 IF NOT DA THEN 970
910 DV=RND(120)*10-5:DS=DS+2*LV*SGN(KP-DS)
920 PUT SPRITE 1,(DS,32+DV),15,DP(DN)+JB
930 IF DE THEN RETURN
940 IF DL THEN 980
950 IF ABS(KP-DS)<(8+LV) THEN LD=51:PUT SPRITE 2,(DS,LD),9,48:DL=-1
960 IF LK THEN 1020
970 RETURN
980 LD=LD+8*LV:IF LD>148 THEN 990 ELSE VPOKE &H1B08,LD:GOTO 1010
990 IF ABS(VPEEK(&H1B09)-KP)<12 THEN 1060
1000 DL=0:PUT SPRITE 2,(0,0),0,63
1010 IF NOT LK THEN RETURN
1020 KT=KT-48:IF KT<32 THEN 1030 ELSE VPOKE &H1B0C,KT:SOUND 10,KT\8:GOTO 1050
1030 LK=0:SOUND 7,56:SOUND 10,0:IF ABS(VPEEK(&H1B0D)-DS)<9 THEN 1070
1040 PUT SPRITE 3,(0,0),0,63:STRIG(JV) ON
1050 RETURN
1060 PUT SPRITE 2,(0,0),0,63:PUT SPRITE 3,(0,0),0,63:DL=0:DE=-1:RETURN
1070 PUT SPRITE 2,(0,0),0,63:PUT SPRITE 3,(0,0),0,63:DL=0:DA=0:DD=-1:RETURN
1080 STRIG(JV) OFF:KT=132:PUT SPRITE 3,(KP,KT),7,11:SOUND 7,28:SOUND 6,10:LK=-1:
RETURN
1090 KEY OFF:SCREEN1,3,0:WIDTH32:GOSUB 1320
```

```
1100 IA=0:IE=15:GOSUB 1980:IA=32:IE=223:GOSUB 1990:GOSUB 1690:RESTORE 1370:COLOR
 1,1,1:WIDTH32:BASE(7)=&H1800
1110 FOR WI=0 TO 3:VDP(1)=162:BASE(5)=WI*&H400:SCREEN1:VDP(1)=162
1120 FOR I=1 TO 20:K=RND(128)*512:J=RND(64)*500:IF J<250 THEN VPOKE BASE(5)+K,12
1 ELSE VPOKE BASE(5)+K,122
1130 NEXT
1140 LOCATE 9,15:PRINT"cbbaaaaaaaabbc":READ A:LOCATE 0,16:PRINT"ccba`gfed"+A+"de
fg`abcc"
1150 FOR I=17 TO 20:READ A:LOCATE 0,I:PRINTA:NEXT
1160 LOCATE 0,21:PRINT"♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀♀":LOCATE 0,22:PRINT"♀♀♀♀♀
♀ :;< 0               xx ♀♀♀♀♀♀"+CHR$(30)+CHR$(30)
1170 LOCATE 0,23:PRINT"♀♀♀♀♀♀♀♀♀♀♀☺☻♥♦♣♠•◘○◙♂♀♀♀♀♀♀♀♀♀♀"+CHR$(30)+CHR$(30):VPOKE
BASE(5)+767,12
1180 REM 11*[g^M][g[][g^(][g^"][g^:][g'][g;][g9][g^(][g0][g^)][gM]9*[g^M]
1190 LOCATE 21,2:PRINT"@ABCDE":LOCATE 21,11:PRINT"PQRSTU"
1200 LOCATE 20,3:PRINT"FttttttG":FOR I=4 TO 9:LOCATE 19,I:PRINT CHR$(I+84)+"tttt
tttt"+CHR$(I+68):NEXT:LOCATE 20,10:PRINT"OttttttN"
1210 LOCATE 25,3:PRINTCHR$((32+WI)*4)+CHR$((32+WI)*4+2):LOCATE 25,4:PRINTCHR$((3
2+WI)*4+1)+CHR$((32+WI)*4+3)
1220 LOCATE 22,9:PRINTCHR$((32+WI)*4)+CHR$((32+WI)*4+2):LOCATE 22,10:PRINTCHR$((
32+WI)*4+1)+CHR$((32+WI)*4+3)
1230 LOCATE 26,8:PRINTCHR$((36+WI)*4)+CHR$((36+WI)*4+2):LOCATE 26,9:PRINTCHR$((3
6+WI)*4+1)+CHR$((36+WI)*4+3)
1240 LOCATE 24,6:PRINTCHR$((36+WI)*4)+CHR$((36+WI)*4+2):LOCATE 24,7:PRINTCHR$((3
6+WI)*4+1)+CHR$((36+WI)*4+3)
1250 LOCATE 20,4:PRINTCHR$((40+4*WI)*4)+CHR$((40+4*WI)*4+2):LOCATE 20,5:PRINTCHR
$((40+4*WI)*4+1)+CHR$((40+4*WI)*4+3)
1260 LOCATE 20,6:PRINTCHR$((42+4*WI)*4)+CHR$((42+4*WI)*4+2):LOCATE 20,7:PRINTCHR
$((42+4*WI)*4+1)+CHR$((42+4*WI)*4+3)
1270 LOCATE 22,4:PRINTCHR$((41+4*WI)*4)+CHR$((41+4*WI)*4+2):LOCATE 22,5:PRINTCHR
$((41+4*WI)*4+1)+CHR$((41+4*WI)*4+3)
1280 LOCATE 22,6:PRINTCHR$((43+4*WI)*4)+CHR$((43+4*WI)*4+2):LOCATE 22,7:PRINTCHR
$((43+4*WI)*4+1)+CHR$((43+4*WI)*4+3)
1290 NEXT WI
1300 VDP(1)=162:VDP(4)=5:VDP(6)=7:RESTORE 1410:FOR I=0 TO 27:READ J:VPOKE &H2000
+I,J:NEXT
1310 FOR I=0 TO 4:READ DP(I):NEXT:RETURN
1320 WIDTH28:PRINT"A L I E N   A T T A C K"
1330 PRINT"-----------------------"
1340 LOCATE 0,3:PRINT"Das nachfolgende Programm":PRINT"ist ein reines MSX BASIC"
:PRINT"Programm und enhält keine":PRINT"Maschinensprache Elemente."
1350 PRINT:PRINT"Der Bildschirm wird gleich":PRINT"für einige Sekunden schwarz.D
ies hat jedoch keine":PRINT"Bedeutung."
1360 RETURN
1370 DATA ddprddddddddpr,ddddddlndddddddlnddddddddlndddddd,dddddmoddddddddmoddddd
ddmodddddd,dhjddddddhjddddddhjddddddddhjd,dikddddddikddddddikddddddddikd
1380 DATA ddddprdddddddd,ddddprddddlndddddddlndprdddddd,ddddddddddmoddddddddm
odddddddd,ddddddhjddddddhjddddddhjdddddd,ddddddikddddddikddddddikdddddd
1390 DATA dprddddddprddd,dprddlnddddddddlnddddddddlndddprd,ddddmoddddddddmoddddd
ddmodddddd,dddddddddhjdddddddddhjdddddddd,dddddddddikddddddddikdddddddd
1400 DATA ddddddprdddddd,ddlndddprdlnddddddlnddddprddlndd,ddmoddddddmoddddddmodd
ddddddmodd,ddddhjdddddddhjdddddddhjdddddd,dddddikddddddikddddddddikdddddd
1410 DATA 252,252,0,0,0,0,241,241,81,81,81,81,225,94,94,161,165,165,165,165,165,
165,165,165,165,165,165,165
1420 DATA 12,16,20,24,28
1430 REM SPRITES
1440 DATA 0000205088C4A2B1,AAA7A0A0A0A0C080,000008182868A868,A828282828 2F203F,00
001F207FC0A09F,403F000000FF00FF,0000FF00FF0000F8,04E2120A0CF932E7
1450 DATA 0000C02090492612,0914264990204080,00003C4890204080,008040209048241E,00
3F40FFA0A0A0A0,BF80FFA0A0A0C080,00E163A5A5A5A5A5,A5A5A5A5A5A5C687
1460 DATA 0002060A0C0A060A,0A0A0A0A0ACA8C08,000000000001F207F,507F407F505F607F,00
00000000E163A4,A7A5C58505E54684,00000000007884F2,0A0A0A0A0A0A0C08
1470 DATA 003F40FFA0A0A0A0,BF80FFA0A0A0C080,00E060A0A0A3A4AF,A0A0A0A0A0A0C080,00
2060A0A0FE04F9,A0A0A0A0A0A0C080,00040C14147F80FF,1414141414141810
```

```
1480 DATA 0000000000CF903F,000F103F282F303F,000000000E364AF,AAAAAAAAAABCC8F,00
00010202FA12E2,0202020202FA13E2,0080848C94A8D0A0,40A0D0A8948A0602
1490 DATA 0001010111113945,BBC5BBD5ABB5C986,0080808088889CA2,DDA3DDABD5AD9361,00
00000000000000,0001000100010001,0000000000000000,8000800080008000
1500 DATA 412211082552 8E91,8E5224558A442810,040810A04894E212,E2944854A2442810,01
0111692552885 1,8C5224552A542800,0000102C48942214,62944854A8542800
1510 DATA 0100010A15220911,09550E5522140800,0080002050882010,2054E05488502000,01
01116925528851,8C5224552A542800,0000102C48942214,62944854A8542800
1520 DATA 0081C2A59069552A,170D02042A542000,00C1A3458916AAD4,68B04020542A0400,00
01022251E9140B,060102040A102800,008040448A17A850,E080402050081400
1530 DATA 00010202013955AA,070102150804 0A00,00C0A040801CAAD5,608040A810205000,00
01022251E9140B,060102040A102800,008040448A17A850,E080402050081400
1540 DATA 8041261918252249,5522241918264180,01826418982444AA,9244A41898648201,00
0023191805222 8,3422041918230000,0000C4189820442C,1444A01898C40000
1550 DATA 00000010A150202,140B040804000000,00000201020D028,4040A85080000000,00
00231918052228,3422041918230000,0000C4189820442C,1444A01898C40000
1560 DATA 0102050A152A542A,1128458A11201008,8040A050A8542A54,8814A25188040810,01
02050A142A5128,142A452A11081000,8040A05028548A14,2854A25488100800
1570 DATA 0102040A1128542A,152A152A15080000,8040205088142A54,A854A854A8100000,01
02050A142A5128,142A452A11081000,8040A05028548A14,2854A25488100800
1580 DATA 0002040928142201,2214280904020000,0040201094284480,4428941020400000,00
08080809140201,0214090808080000,0010109010284080,4028109010100000
1590 DATA 0010205009040201,0204095020100000,0008048A10204080,4020108A04080000,00
08080809140201,0214090808080000,0010109010284080,4028109010100000
1600 DATA 0F10274893A4A9AA,AAA9A4934827100F,F008E412C9259555,559525C912E408F0,00
0007081324292A,2A29241308070000,0000E010C8249454,549424C810E00000
1610 DATA 000000000304090A,0A09040300000000,00000000C0209050,509020C000000000,00
00000000000102,0201000000000000,0000000000008040,4080000000000000
1620 DATA 1000450111113144,ABC52BC5AB14C986,00A0808288009CA2,C9A3912BD1A59360,10
00441104112144,AB442A40AA04C986,00A4002288001CA2,49A3102951A51200
1630 DATA 0000401100110044,0B042A00AA004984,0084000208010A2,08A2100950251200,00
00001100000044,0000080002000900,0000000000001000,0082000040041000
1640 DATA 003F40FFA0A0A0BF,80FFA0A0A0BFC0FF,00F020C0000000F0,30D0505050D060C0,00
3F40FFA0A0A0BF,80FFA0A0A0A0C080,00F030D0505050D0,50D0505050506040
1650 DATA 00205088C4A2B1AA,A7A0A0A0A0A0C080,0008182868A868A8,2828282828283020,00
3F40FFA0A0A0FF,81FEA0A0A0BFC0FF,00F020C000000080,000000000F020C0
1660 DATA 003F40FFA0A0A0A0,A0A0A0A0A0BFC0FF,00F030D050505050,505050505 0D060C0,00
2050908848442 4,2211110A0A040407,0008142424484890,9020204040808080
1670 DATA 003F40FFA0A0A0FF,81FEA0A0A0BFC0FF,00F020C000000080,000000000F020C0,00
3F40FFA0A0A0BF,80FFA8A4A2A1C080,00C0A010B05050E0,C00080402010A040
1680 DATA 0000000103060505,0201000000000000,00000080C0E0E0E0,C080000000000000
1690 RESTORE 1440
1700 FOR I=0 TO 195:READ C$:FOR K=0 TO 7:VPOKE 8*I+K+&H3800,VAL("&H0"+MID$(C$,2*
K+1,2)):NEXT K,I:RETURN
1710 REM CHARAKTER
1720 DATA 0000000000000000,003058A4A4583000,00BDA5BD8585BD00,00EF28EF2121EF00,00
22221414080800,000404F49494F600,008080AFCFA8AF00,0003025362424300
1730 DATA 0080409E5E509E00,000404F58685F500,0000007A7B427A00,0000008000000000,00
00000000000000,302824242820E0C0,3C243C2424E4DC18,105438EE38541000
1740 DATA 0000000000000000,2020202000002000,5050500000000000,5050F850F8505000,20
78A07028F02000,C0C8102040981800,40A040A890986000,1020400000000000
1750 DATA 1020404040201000,4020101010204000,20A8702070A82000,002020F820200000,00
00000000202040,0000007800000000,0000000000606000,0000081020408000
1760 DATA 0018242424241800,0010301010103800,0018240408103C00,0018240804241800,00
2828283C080800,003C203804043800,0018203824241800,003C040810101000
1770 DATA 0018241824241800,0018242410041800,00F880F80A0AF800,000000DE1212DE00,00
0000AECE888E00,0000F800F8000000,C06030183060C000,7088081020002000
1780 DATA 00000000030F3F7F,00073FFFFFFFFFFF,3FFFFFFFFFFFFFFF,FCFFFFFFFFFFFFFF,00
E0FCFFFFFFFFFF,00000000C0F0FCFE,0003070F1F3F7F7F,00C0E0F0F8FCFEFE
1790 DATA 0080C0C0E0E0F0F0,F8F8FCFCFCFEFEFE,FEFEFFFFFFFFFFFF,FFFFFFFFFFFFFEFE,FE
FEFEFCFCFCF8F8,F0F0E0E0C0C08000,FEFEFCF8F0E0C000,7F7F3F1F0F070300
1800 DATA 7F3F0F0300000000,FFFFFFFFFF3F0700,FFFFFFFFFFFFFF3F,FFFFFFFFFFFFFFFC,FF
FFFFFFFFFFCE000,FEFCF0C000000000,8888888850502000,888888A8A8D88800
1810 DATA 0001030307070F0F,1F1F3F3F3F7F7F7F,7F7FFFFFFFFFFFFF,FFFFFFFFFFFF7F7F,7F
```

```
7F7F3F3F3F1F1F,0F0F070703030100,2050880000000000,000000000000F800
1820 DATA 00000000FFFFFFFF,0000000000FFFFFF,000000000000FFFF,00000000000000FF,FF
FFFFFFFFFFFFFF,00FFFFFFFFFFFFFF,0000FFFFFFFFFFFF,000000FFFFFFFFFF
1830 DATA 0000030E382E332E,352E754E6DDAB5DA,0000C0701C74CCB4,6CB46EB256BB5DAB,00
000000030C0B0A,1A1234252A190700,000000008060A0A0,B0905848A830C000
1840 DATA 0102050D0A0D0700,0000000000000000,8040A030D050E000,0000000000000000,FF
FFFFFFFFFFFFFF,0004004000010800,0000480000401000,8200000022000400
1850 DATA 001010547C7C5400,0000103810000000,0000000008000000,1820204020201800,20
20200020202000,C02020102020C000,40A8100000000000,00002050F8000000
1860 DATA 0F1F306043C78C19,32333130180C0703,C0F0180CC6E333B3,33E3C6060C18F0E0,00
000F1F3060C3C6,C6C6C361301C0F03,60380C86C36333B3,B333E2C60C18F0E0
1870 DATA 070F18306063C7CC,CDCCC76330180F03,C0E030180C8CCC4C,9831E3C2060CF8F0,07
0F18306347CCCD,CDCCC6C361301C06,C0F0380C86C36363,63C3060CF8F00000
1880 DATA 061C3061C3C6CCCD,CDCC476330180F07,0000F0F80C06C363,6363C3860C38F0C0,03
070C1830313332,198CC74360301F0F,E0F0180C06C6E333,B333E3C60C18F0C0
1890 DATA 030F1C3061C3C6C6,C6C360301F0F0000,E0F0180CC6E233B3,B33363C3860C3860,03
0F183063C7CCCD,CCC7636030180F07,F0F80C06C2E33198,4CCC8C0C1830E0C0
1900 DATA 0000000000000001,03060606060C0C0C,00000000071F78E0,800F1F30316366CC,3F
0F0080C0FE3F01,01F0F80C86C66676,00E0F0181C0C0C86,C6C3C36363636363
1910 DATA 0C0C0C0606030301,0100000000000000,CCC66361303C0F83,80C0F87F07000000,8C
18F0E1071EF8E0,00030FFCF0000000,63C3C38686060C3C,F0C0000000000000
1920 DATA 0000000000000001,070E0C0C0C181818,00000000071F78E0,830F3C306163C6CC,00
000000E0FE1F01,81E0781C86C66636,0000000000000080,80C0C06060303030
1930 DATA 181818180C0C0606,0303010100000000,CDCCCC6663313018,1E078180C0F83F07,36
E6C6860CF8F001,07FFF8000001FFFE,38306161636363C3,8606061C78F08000
1940 DATA 000000000000030F,3C30606161C3C3C6,0000000F3FF0C000,071F78E0870F1831,00
0000E0FE1F0301,C1F03C0C86C66333,0000000000000080,80C0C06060303030
1950 DATA C6C6C6C6C6C3C363,61303038180F0700,6E666361301F0F80,80FC7F030100F0FC,33
66C68C0CF8F001,071EF8E000000000,30303060606060C0,8000000000000000
1960 DATA 00010F1E38606061,C3C6C6C686860C1C,7FFF8000001FFFE0,800F1F306163676C,E0
FC1F030181E078,180C8CC6663333B3,000000008080C0C0,6060303018181818
1970 DATA 0C0C0C0606030301,0100000000000000,6C666361381E0F81,80F87F0700000000,33
63C6860C3CF0C1,071EF8E000000000,18181830303070E0,8000000000000000
1980 RESTORE 1720
1990 FOR I=IA TO IE:READ C$:FOR K=0 TO 7:VPOKE 8*I+K+&H2800,VAL("&H0"+MID$(C$,2*
K+1,2)):NEXT K,I:RETURN
2000 INTERVAL OFF:STRIG(JV) OFF:SOUND 8,0:SOUND 9,0:SOUND 10,0
2010 BASE(5)=&H1800:BASE(7)=0
2020 SCREEN0:COLOR 15,4,4:RETURN
2030 GOSUB 2000:END
2040 GOSUB 2000:PRINT"Fehler in Zeile ";ERL
2050 ERROR ERR
2060 END
```

# Tips und Tricks CPC-464: Reverse On/Off



Leider gibt es beim CPC keinen Befehl, der auf den „Reverse-Modus" umschaltet. Wer nicht die umständliche Methode der Paper/Pen-Auswechslung vornehmen möchte, kann folgende Systemroutine aufrufen!

CALL &BB9C:rem Vertauscht Paper und Pen Farbe

# BTX setzt sich durch

Seit Mai/Juni 1984 bietet die Post den uneingeschränkten bundesweiten Bildschirmtext an. Jeder Bundesbürger kann sich seit diesem Zeitpunkt an dieses Datennetz anschließen lassen.
Der folgende Artikel soll die Möglichkeiten und Kosten für den Teilnehmer aufzeigen.

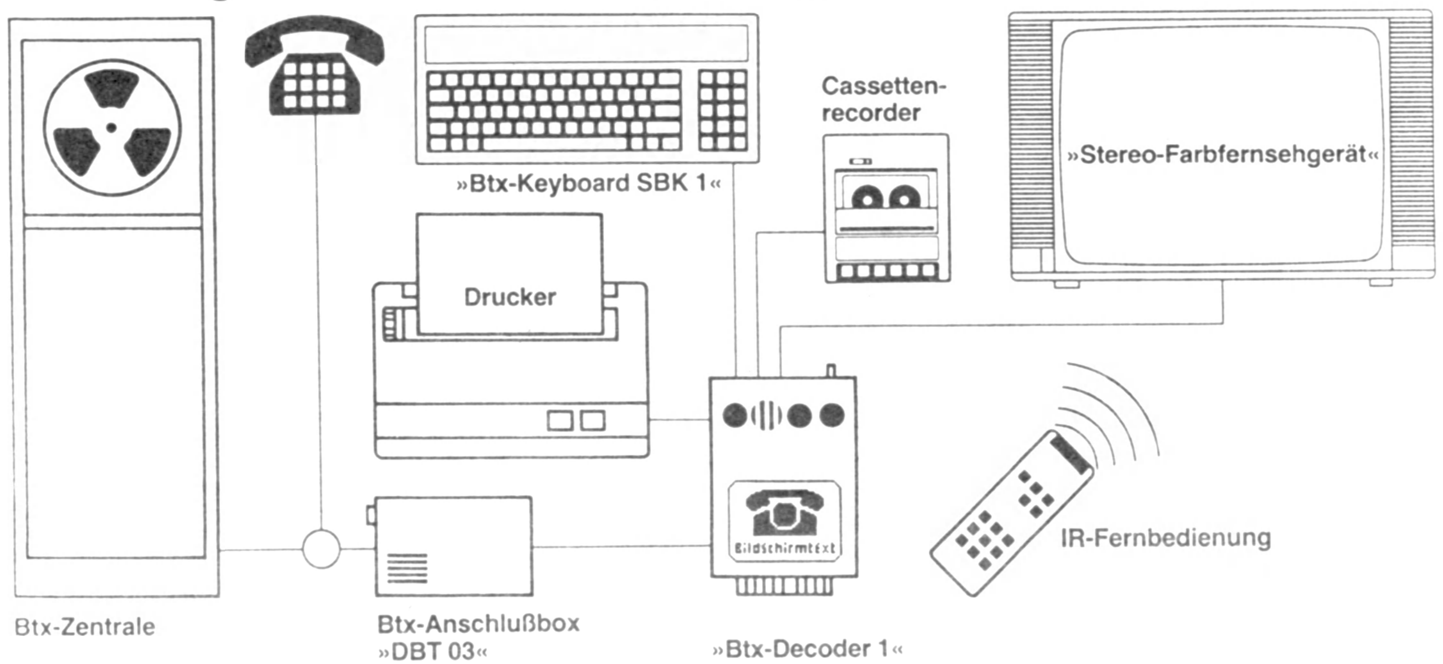
**Was verbirgt sich hinter dem Begriff Bildschirmtext?**
Bildschirmtext ist ein neuer Fernmeldedienst der Deutschen Bundespost. Neben altbekannten Kommunikationstechniken wie Telefon und Telex und neueren wie Telefax und Teletex bietet die Post seit 1984 den Bildschirmtext-Dienst an.
Durch die Koppelung von Fernsehgerät und Telefon sowie der Verbindung zu einem Computer werden der Empfang und das Senden von Informationen über die Telefonleitung möglich. Über die Kommunikation mit den BTX-Zentralen (Computern) der Post hinaus, können auch EDV-Anlagen von Unternehmen, Organisationen oder Institutionen angewählt werden. Durch diesen „Rechnerverbund" werden die Dialogmöglichkeiten nahezu unbegrenzt erweitert. BTX liefert Textinformationen und Grafiken in vielen Farben auf den Bildschirm eines Fernsehapparates. Man besitzt die Möglichkeit aus inzwischen mehreren Hunderttausend Bildschirmtext-Seiten eine Auswahl zu treffen. Zum Beispiel können Sie den neuesten Wetterbericht oder aktuelle Reiseangebote abrufen. Und da BTX auch dialogfähig ist, können Sie auch Überweisungen von Ihrem Bankkonto veranlassen, oder eine Bestellung bei Ihrem Versandhaus aufgeben.

**Aufbau des BTX-Systems**
Die Deutsche Bundespost stellt das Telefonnetz, das BTX-System sowie die Vermittlungsstellen zur Verfügung. Die Informationen werden von über 3000 Anbietern in das BTX-System eingegeben und in den BTX-Vermittlungsstellen der Post gespeichert.
Datenverarbeitungsanlagen und Rechenzentren der verschiedensten Anbieter (z. B. Banken, Versicherungen, Versandhäuser, Reiseunternehmen, Verlage) werden über das Datex-Netz der Post mit den BTX-Vermittlungsstellen verbunden.
BTX wurde erstmals im September 1983 zur internationalen Funkausstellung eingeführt; wobei mit dem CEPT-Standard ein noch größerer Vorrat an Zeichen und Farben sowie eine verfeinerte Wiedergabe grafischer Darstellungen durch höhere Bildauflösung ermöglicht wurde. Seit dem Sommer 1984 ist eine neue Systemtechnik im Einsatz, die die technische Basis für den weiteren Ausbau des BTX-Dienstes bildet.

**BTX für jedermann?**
Nach mehrjähriger öffentlicher Erprobung ist deutlich geworden, daß BTX als neues Informationsmedium und insbesondere als neuer Kommunikationsweg für die Vorbereitung und Erledigung privater Geschäfte und allgemeiner Informationsbedürfnisse von der überwiegenden Mehrheit der privaten Teilnehmer akzeptiert wird. Zur Zeit sind etwa 20 000 private Teilnehmer an diesem System angeschlossen. Diese Zahl dürfte sich in den kommenden Monaten und Jahren noch um ein vielfaches steigern, da inzwischen die Kosten für das Betreiben eines BTX-Decoders sehr günstig sind. Immer mehr Modem-Freaks steigen auf BTX um, da die Möglichkeiten weitaus größer und die Telefonkosten wesentlich niedriger sind, als die Verbindung mit fernen Mailboxen.
BTX bietet sich dem privaten Nutzer geradezu an, da er seine Aktivitäten bequem vom Wohnzimmer aus tätigen kann. Durch die 24stündige Verfügbarkeit ist er weder an Ladenschluß- noch an andere Öffnungszeiten gebunden.
Folgende BTX-Möglichkeiten werden in erster Linie von privaten Nutzern in Anspruch genommen:

1. Angebote von Waren und Dienstleistungen aus allen Wirtschaftsbereichen
2. Aktuelle Nachrichten

## Verbindungsschema und Aufbau des Mediums »Bildschirmtext«



(Presse) und aktuelle Informationen mit Servicecharakter
3. Interaktive Dienste, insbesondere Waren- und Informationsbestellungen
4. Kommunikation von BTX-Teilnehmern untereinander.

**Wie funktioniert BTX?**

Der private BTX-Teilnehmer braucht nicht viel: Einen Farbfernseher mit BTX-Decoder und Fernbedienung, ein Telefon und eine BTX-Anschlußbox (Modem).

Zu beachten ist, daß BTX-Decoder nicht an alle Fernseher angeschlossen werden können, man sollte sich deshalb vor dem Kauf mit dem Händler absprechen. BTX-Decoder gibt es zur Zeit in der Preislage zwischen 700 und 2000 DM. Sicherlich ist auch hier in nächster Zeit ein Preisrückgang zu erwarten.

Der BTX-Decoder speichert die empfangenen Textinformationen und wandelt sie in stehende Fernsehbilder, sogenannte Bildschirmtext-„Seiten", um. Die BTX-Abschlußbox verbindet den BTX-Decoder mit dem Telefonanschluß. Die Übermittlungsgeschwindigkeit für BTX-Seiten beträgt 1200 bit pro Sekunde (1200 Baud), also 4 mal so schnell wie eine übliche Mailbox. Die Übertragung von eigenen Seiten an die BTX-Zentrale läuft dagegen wesentlich langsamer ab, hier wird aus Sicherheitsgründen nur mit 75 Baud gearbeitet. Für den Dialog mit dem BTX-System benutzen Sie entweder die Fernbedienung Ihres Fernsehers oder die BTX-Tastatur des Decoders. Damit lassen sich Buchungen und Bestellungen durchführen. Auf jeder BTX-Seite finden Sie einen Hinweis mit welchen Tasten Sie die nächste Seite abrufen können. Der Anbieter von BTX-Seiten besitzt die Möglichkeit diese mit einem Betrag von 0,00 DM bis 9,99 DM zu belegen. Dieser Preis muß jedoch in einer vorhergehenden Seite angekündigt werden, so daß der BTX-Teilnehmer die Wahl besitzt, diese Seite aufzurufen oder zu überspringen. Wird eine solche Seite angewählt, so wird der entsprechende Betrag mit Ihrer Telefonrechnung eingezogen.

Solange Sie Bildschirmtext benutzen, ist Ihr Telefonanschluß „besetzt". Möchten Sie trotzdem telefonisch erreichbar sein oder gleichzeitig telefonieren, so benötigen Sie einen Doppelanschluß.

Die abgerufenen BTX-Seiten lassen sich mit einem gewöhnlichen Kassettenrecorder oder seit neuestem auch mit einer Diskettenstation abspeichern. Die so gespei-

**Beispiele für Anwendungen**

| Anwendungen (Beispiele) | Mögl. Informationsanbieter (Auswahl) |
|---|---|
| Politische Nachrichten | Tageszeitungen, Presseagenturen |
| Wetterbericht/Reisewetterbericht | Deutscher Wetterdienst |
| Sport | Tageszeitungen, Sportzeitungen |
| Lotto/Toto | Tageszeitungen, Klassenlotterien |
| Urlaubsreisen/Zimmernachweis | Reiseveranstalter, Reisebüros, Hotels |
| Fahrplanauskunft | DB, Verkehrs- und Fluggesellschaften |
| Theater- und Konzertprogramme | Veranstalter, Kommunen |
| Lokaler Veranstaltungskalender | TZ, Vereine, Kommunen, Parteien |
| Bestseller/Neuveröffentlichungen | Verlage, Buchclubs |
| Verkaufsangebote | Kaufhäuser, Versandhandel, Cash & Carry |
| Immobilien | Makler |
| Stellenangebote | Arbeitsämter, Firmen |
| Bestellungen | Versandhandel, Buchclubs, Theaterkassen |
| Buchungen | Reiseveranstalter, Reisebüros |
| Überweisungen | Banken, Sparkassen, Postgiroämter |
| Schadensmeldungen | Versicherungen |
| Finanzierung | Bausparkassen, Banken |
| Steuererklärung | Steuerberater, Finanzamt |
| Aus- und Weiterbildung | Fachverlage, Fernlehrinstitute |
| Tests | Psychologische Institute |
| Spiele | Unternehmen der Unterhaltungsbranche |

**Fortsetzung S. 28**

cherten Seiten können jederzeit auf Ihren Bildschirm gebracht werden, ohne das Gebühren dafür entrichtet werden müssen. Wie auch bei den Heimcomputern läßt sich auch ein Drucker anschließen, der BTX-Zeiten ausgeben kann.

Ein besonders interessanter Dienst sind die „Persönlichen Mitteilungen". Dieser Dienst ist ähnlich aufgebaut, wie bei den „Mailbox-Briefkästen", die immer mehr Freunde finden. Bei BTX hat jeder Teilnehmer einen eigenen elektronischen Briefkasten. Nach dem Beginn eines Dialogs finden Sie auf der Begrüßungsseite einen Hinweis, ob für Sie eine Mitteilung vorliegt. Für Ihre eigene Mitteilung an einen anderen BTX-Teilnehmer gibt es zwei Möglichkeiten:

**1. Möglichkeit**
Anwählen einer Art „Formular", welches Sie mit Zahlenangaben für Bestellungen, Datum u. a. ergänzen können.

**2. Möglichkeit**
Aufrufen einer Leerseite, in die mit einer BTX-Tastatur beliebiger Text eingegeben werden kann. Bestimmen des Empfängers durch Eingabe seiner BTX-Teilnehmernummer.

**Was kostet Bildschirmtext?**
Trotz umfangreicher Werbe-Aktionen der Post sind die eigentlichen Kosten nur sehr wenigen bekannt. Der Vorwurf BTX sei für den privaten Gebrauch zu teuer, wird jedoch zu Unrecht erhoben. Der Preis für Fernsehgerät mit BTX-Decoder liegt etwa 800-1500 DM über dem eines einfachen Fernsehempfängers. Wie schon erwähnt wird dies in den nächsten Monaten sicher noch günstiger werden. Für die Leistungen im BTX-Dienst erhebt die Post folgende Gebühren:

**An alle BTX-Teilnehmer**
**Die CPU beabsichtigt auch weiterhin von BTX zu berichten. Haben Sie Erfahrung mit BTX so können Sie mit einem Artikel zum Inhalt der CPU beitragen. Das Honorar beträgt 120 DM pro abgedruckte Seite.**
**Wir erwarten gespannt auf Ihre Beiträge!**

**Gebühren für Btx-Teilnehmer**

| Leistung | Betrag | |
|---|---|---|
| – Anschließungs- oder Änderungsgebühr | DM 65,00 | einmalig |
| – Monatliche Gebühr je Teilnehmerkennung | DM 8,00 | |
| – Telefongebühr | Orts-/Nahgesprächsgebühr[1]) | |
| – Einrichten von Mitbenutzern | DM 0,05 | pro Tag und Mitbenutzern[3]) |
| – Absenden einer Mitteilung | DM 0,40 | pro Seite[2]) |
| – Speichern einer abgerufenen Mitteilung | DM 0,015 | pro Tag u. Seite[2]) |
| – Abruf aus anderen Regionalbereichen | DM 0,02 | pro Seite[2])[3]) |

# Btx – neue Applikation für den Texas Instruments Professional-Computer

**Der Btx-Software-Decoder des Dortmunder System- und Softwarehauses mbp erhält die FTZ-Zulassung auf einem Texas Instruments Professional-Computer**

Mit der Erteilung einer allgemeinen Zulassung für den vom Mathematischen Beratungs- und Programmierdienst (mbp) entwickelten Software-Decoder auf dem PC von Texas Instruments durch das Zentralamt für Zulassungen im Fernmeldewesen wurde ein großer Fortschritt für die Anwendung von PCs im Rahmen von Bildschirmtext erzielt.
Zum erstenmal in ihrer über 30jährigen Zulassungspraxis hat das Darmstädter FTZ mit der FTZ-Nummer 0 3841D einer Software ihren Segen gegeben. Der Btx-Software-Decoder des Dortmunder System- und Softwarehauses mbp hat die FTZ-Zulassung auf einem TI-PC.
Die entscheidende Voraussetzung, Bildschirmtext wirklich zum Kommunikationsmedium der Zukunft zu machen, besteht in der Möglichkeit, Btx voll in den vorhandenen Informationskreislauf zu integrieren. Erst wenn der professionelle Anwender über ein Terminal nicht nur seine üblichen PC-Applikationen, sondern auch seine Kommunikationsanforderungen vernünftig bearbeiten kann, lassen sich die Vorteile von Btx für ihn angemessen nutzen.
Realisierbar wird diese Lösung durch einen Btx-Software-Decoder, wie der von mbp entwickelte. Die zur Weiterverarbeitung im Mikrocomputer ohnehin erforderliche Decodierungs-Software wurde durch einen Displayteil ergänzt, so daß die Btx-Seiten auf dem am Rechner ohnehin vorhandenen Sichtgerät dargestellt werden können.
Dabei werden die in den Btx-Seiten enthaltenen Nutzinformationen den Benutzerprogrammen, unabhängig vom verwendeten Btx-Standard, für eine Weiterverarbeitung zur Verfügung gestellt. Das Prinzip der vollständigen Software-Integration ermöglicht so dem Anwender eine gegenüber Hardware-Lösungen höhere Flexibilität in der Realisierung anforderungsgerechter Kommunikationskonzepte.
Die Zulassung, die den Betrieb eines Professional-Computers als Teilnehmer-Terminal gestattet, sieht folgende Konfiguration als Minimum vor:
Texas Instruments (Portable) Professional-Computer
- ein Floppy-Laufwerk
- 256-KByte-Speicher
- TI-Farbgrafik (3 Ebenen)
- Kommunikationsmodul (zum Anschluß an 1200/75 Modem)
- VTX 84 RGB-Adapter (lieferbar 2. Quartal '85)
- MS-DOS 2.11
- mbp Btx-Software Rel. 1.5.

Der Einsatz als Teilnehmer-Terminal gestattet es, die Bildschirmtext-Seiten auf dem Monitor des TI-PC darzustellen. Ein RGB-Monitor kann angeschlossen werden. Btx-Seiten können auf den angeschlossenen Massenspeichern (Floppy- oder Winchester-Disk) gespeichert und im Off-line-Modus ohne Bildschirmtext-Zentrale aufgerufen werden.

# Zeicheneditor für ATARI (48k RAM) + Diskettenstation

Der „ZEICHENEDITOR“ ist ein Utility, das dem Anwender erlaubt, den Zeichensatz des ATARI umzugestalten und ihn in seinen eigenen Programmen zu nutzen.

Der „ZEICHENEDITOR“ meldet sich auf dem Bildschirm mit einem Menue, das alle Möglichkeiten anzeigt. Zur Auswahl der Funktionen ist der Anfangsbuchstabe der Funktion mit der Tastatur einzugeben.

Das Zeichen wird mit Hilfe des Joysticks editiert. Unter dem Menuepunkt „SAVEN“ findet man auch die Möglichkeit, den editierten Zeichensatz als eigenes Programm abzuspeichern. Es muß mit ENTER eingeladen werden.
Die Funktion „BANNER“ setzt einen Drucker voraus. Mit dieser Funktion ist es möglich, große Schriftbänder zu drucken.

Beim Abtippen empfiehlt es sich, das Programm vorher abzuspeichern, da es mehrere Maschinenunterprogramme beinhaltet, die bei fehlerhaftem Abtippen zum Absturz führen können.
Ein Drucker ist nur für die Funktion „BANNER“ erforderlich.

```
100 REM ZEICHENEDITOR
110 REM ---------------
120 REM Jens Schuppener
130 REM Denningerstr.206
140 REM 8000 Muenchen 81
150 REM Tel.: 089/93 90 88
160 REM
170 REM *****************
180 REM
190 POKE 106,152:GRAPHICS 17
200 DIM A$(255),TRANS$(42),CURS$(75),INTER$(20),VBV$(88),CLEAR$(31),SPLIT$(112)
210 DIM Q(8,8),W(8),DIR$(20)
220 DIM INVERS$(44),DREHEN$(81),SPIEGELN$(54),X$(40)
230 ? #6;"******************"
240 ? #6;"*  zeicheneditor  *"
250 ? #6;"******************"
260 ? #6:? #6
270 ? #6;"1985 JENS SCHUPPENER"
280 ? #6:? #6
290 ? #6;"bitte warten......"
300 ? #6:? #6;"LADE CHARACTERSATZ"
310 ? #6;"VOM rom INS ram"
320 ? #6:? #6:? #6:? #6
330 ? #6;"   homecomputer '85"
340 REM  CHARACTERMAP INS RAM
350 REM --------------------------
360 CHBASNEU=156*256
370 CHBASOLD=PEEK(756)*256
380 GOSUB 4700:REM VBI
390 GOSUB 4950:A=USR(ADR(CLEAR$))
400 GOSUB 4480:A=USR(ADR(TRANS$))
410 GOSUB 5030:A=USR(ADR(CURS$),0,4)
420 GOSUB 5400:REM DLI INTERRUPT
430 GOSUB 5240:REM SPLIT
440 GOSUB 5150:REM INVERS
450 GOSUB 4570:REM DREHEN
460 GOSUB 4850:REM SPIEGELN
470 GOSUB 5730:REM TITEL
480 REM DISPLAYLIST
490 REM ------------
500 GRAPHICS 0
510 POKE 752,255:REM CURSOR AUS
520 POKE 709,18:POKE 710,24
530 DLT=38912:REM $9800
```



```
540 SCREENBASE=37952:REM $9440
550 DATA 112,112
560 DATA 82,00,06
570 DATA 16
580 DATA 66,64,148
590 DATA 2,2,2,2,2,2,2,2,130,2
600 DATA 144,2,2,2,2,16,4,16,5,16,6,16,7
610 DATA 65,00,152,-1
620 ZAEHLER=DLT:RESTORE 550
630 READ D:IF D=-1 THEN 660
640 POKE ZAEHLER,D:ZAEHLER=ZAEHLER+1
650 GOTO 630
660 POKE 560,DLT-INT(DLT/256)*256
670 POKE 561,INT(DLT/256)
680 POKE 512,ADR(INTER$)-INT(ADR(INTER$)/256)*256
690 POKE 513,INT(ADR(INTER$)/256)
700 POKE 54286,192
710 REM  DISPLAY
720 REM ---------
730 ?
740 ? "Atari                      1 OLDCHR"
750 ? "Invert                     2 LADEN"
760 ? "Drehen                     3 SAVEN"
770 ? "Kopieren                   4 BANNER"
780 ? "Loeschen                   5 DIR"
790 ? "Spiegeln"
800 ? "Waehlen"
810 ? "Editieren"
820 POSITION 14,0:FOR I=1 TO 10:? CHR$(160);:NEXT I
830 FOR I=1 TO 8:POSITION 14,I
840 ? CHR$(160);"        ";CHR$(160)
850 NEXT I
860 POSITION 14,9:FOR I=1 TO 10:? CHR$(160);:NEXT I
870 REM  PLAYER MISSLE
880 REM ---------------
890 POKE 559,46:REM DMA EIN
900 POKE 53277,3:REM DMA EIN
910 POKE 623,1:REM PRIOR
920 POKE 53257,3:REM SIZE P1
930 POKE 705,116:REM FARBE P1
940 POKE 53258,3:REM SIZE P2
950 POKE 706,22:REM FARBE P2
960 POKE 53256,0:REM SIZE P0
970 POKE 53249,108:REM P1  X-WERT
980 POKE 53250,108:REM P2  X-WERT
990 POKE 54279,152:PMBASE=152*256:REM PMBASE
1000 P0BASE=PMBASE+512
1010 P1BASE=PMBASE+640
1020 P2BASE=PMBASE+768
1030 DATA 240,144,144,240
1040 RESTORE 1030
1050 FOR I=1712 TO 1712+3:READ D:POKE I,D
1060 NEXT I
1070 REM MUSTER PM3
1080 REM -----------
1090 FOR I=P2BASE+21 TO P2BASE+21+31 STEP 8
1100 POKE I,170:POKE I+1,170
1110 POKE I+2,170:POKE I+3,170
1120 POKE I+4,85:POKE I+5,85
1130 POKE I+6,85:POKE I+7,85
1140 NEXT I
1150 REM  ALLE ZEICHEN
1160 REM -------------
```

```
1170 ZAEHLER=38396:COUNT=0
1180 FOR I=0 TO 127
1190 IF COUNT=32 THEN COUNT=0:ZAEHLER=ZAEHLER+8
1200 POKE ZAEHLER,I
1210 ZAEHLER=ZAEHLER+1
1220 COUNT=COUNT+1
1230 NEXT I
1240 GOSUB 3860:REM CHR IN GRAFIKEN
1250 REM  HAUPTSCHLEIFE
1260 REM ---------------
1270 A=PEEK(764)
1280 IF A=31 THEN GOSUB 1490:GOTO 1270:REM OLDCHR
1290 IF A=30 THEN GOSUB 2890:GOTO 1270:REM LADEN
1300 IF A=26 THEN GOSUB 2340:GOTO 1270:REM SPEICHERN
1310 IF A=24 THEN GOTO 3070:REM BANNER
1320 IF A=29 THEN GOTO 2160:REM DIR
1330 IF A=63 THEN GOSUB 1730:GOTO 1270:REM ATARI
1340 IF A=58 THEN GOSUB 2040:GOTO 1270:REM DREHEN
1350 IF A=13 THEN GOSUB 1800:GOTO 1270:REM INVERS
1360 IF A=5 THEN GOSUB 1850:GOTO 1270:REM KOPIEREN
1370 IF A=0 THEN GOSUB 1960:GOTO 1270:REM LOESCHEN
1380 IF A=62 THEN GOSUB 2100:GOTO 1270:REM SPIEGELN
1390 IF A=46 THEN GOSUB 1420:GOTO 1270:REM WAEHLEN
1400 IF A=42 THEN GOSUB 1550:GOTO 1270:REM EDITIEREN
1410 GOTO 1270
1420 REM  WAEHLEN
1430 REM ---------
1440 POKE 764,255
1450 GOSUB 4050:REM CUSOR WAEHLEN
1460 GOSUB 3750:REM SICHTFELD LADEN
1470 GOSUB 3860:REM CHAR IN GRAFIKEN
1480 RETURN
1490 REM  OLDCHR
1500 REM --------
1510 POKE 764,255
1520 A=USR(ADR(TRANS$))
1530 GOSUB 3750:REM CHR INS FENSTER
1540 RETURN
1550 REM  EDITIEREN
1560 REM -----------
1570 POKE 764,255
1580 WX=108:WY=21
1590 A=USR(ADR(CURS$),WX,WY)
1600 A=STICK(0)
1610 IF PEEK(764)<>255 THEN RETURN
1620 IF STRIG(0)=0 THEN GOSUB 1700:GOTO 1660
1630 IF A=14 AND WY>21 THEN WY=WY-4:GOTO 1680
1640 IF A=13 AND WY<48 THEN WY=WY+4:GOTO 1680
1650 IF A=11 AND WX>108 THEN WX=WX-4:GOTO 1660
1660 IF A=7 AND WX<136 THEN WX=WX+4:GOTO 1680
1670 GOTO 1600
1680 A=USR(ADR(CURS$),WX,WY)
1690 GOTO 1600
1700 X=(WX-106)/4:Y=(WY-21)/4
1710 A=USR(ADR(SPLIT$),X,Y,CHRNR)
1720 RETURN
1730 REM  ATARI
1740 REM -------
1750 POKE 764,255
1760 FOR I=0 TO 7
1770 POKE CHBASNEU+CHRNR*8+I,PEEK(CHBASOLD+CHRNR*6+I)
1780 NEXT I
1790 GOTO 1460
```

Dieses Buch ist Programmier-Lehrbuch, Programmsammlung und Handbuch zum elrad-COBOLD-Computer in einem. Wer einen anderen Computer benutzt, wird einige Angaben auf diesen übertragen müssen – ein Nachteil, den jedes andere nicht systembezogene Lehrbuch auch aufweist. Andererseits bietet aber gerade der Bezug auf einen real vorhandenen Computer und dessen Betriebssystem, das in allen Einzelheiten erläutert wird, den Vorteil der Praxisnähe. Im Prinzip kann jeder 6502-Computer zum Einüben der beschriebenen Programmiertechniken dienen, wenn wenigstens ein Maschinensprache-Monitor vorhanden ist. (Andernfalls können Sie den COBOLD-Monitor übernehmen; allerdings sind dafür schon einige Vorkenntnisse erforderlich.)

Mit dem elrad-COBOLD-Computer steht ein preisgünstiges Selbstbau-System zur Verfügung, das sich gleichermaßen gut als Lerncomputer für den Anfänger wie als Entwicklungssystem oder Zentraleinheit eines Tischcomputers für den fortgeschrittenen Programmierer eignet. So bietet der COBOLD-Computer schon in der Grundversion viele Möglichkeiten, die man bei einem Lerncomputer sonst nicht vorfindet. Der weitere Ausbau ist nicht nur möglich, sondern schon weitgehend vorbereitet, so daß nur noch geringe Kosten entstehen.

Dieses Buch soll Lesern mit sehr unterschiedlichem Hintergrund gerecht werden. Es wendet sich an den Anfänger, der weder auf dem Gebiet der Computertechnik noch auf dem der Mathematik besondere Vorkenntnisse besitzt. Es richtet sich zugleich an den 6502-Experten, der eine Dokumentation des COBOLD-Systems benötigt, um dessen Möglichkeiten auszuschöpfen beziehungsweise zu erweitern.

```
1800 REM  INVERS
1810 REM --------
1820 POKE 764,255
1830 A=USR(ADR(INVERS$),CHRNR)
1840 GOTO 1460
1850 REM  KOPIEREN
1860 REM ----------
1870 POKE 764,255
1880 CHRNR2=CHRNR
1890 GOSUB 4050:REM CURSOR WAEHLEN
1900 GOSUB 3750:REM CHR INS SICHTF.
1910 FOR I=0 TO 7
1920 POKE CHBASNEU+CHRNR2*8+I,PEEK
(CHBASNEU+CHRNR*8+I)
1930 NEXT I
1940 CHRNR=CHRNR2
1950 RETURN
1960 REM  LOESCHEN
1970 REM ----------
1980 POKE 764,255
1990 FOR I=0 TO 7
2000 POKE CHBASNEU+CHRNR*8+I,0
2010 NEXT I
2020 GOSUB 3750:REM CHR INS SICHTF.
2030 RETURN
2040 REM  DREHEN
2050 REM --------
2060 POKE 764,255
2070 A=USR(ADR(DREHEN$),CHRNR)
2080 GOSUB 3750:REM CHR INS SICHTF.
2090 RETURN
2100 REM  SPIEGELN
2110 REM ----------
2120 POKE 764,255
2130 A=USR(ADR(SPIEGELN$),CHRNR)
2140 GOSUB 3750:REM SICHTFELD
2150 RETURN
2160 REM  DIR
2170 REM -----
2180 POKE 53248,0
2190 POKE 53249,0
2200 POKE 53250,0:REM XWERT PL0-2
2210 POKE 764,255
2220 GRAPHICS 0
2230 TRAP 2280
2240 OPEN #1,6,0,"D:*.*"
2250 INPUT #1;X$:? X$;"    ";
2260 INPUT #1;X$:? X$
2270 GOTO 2250
2280 CLOSE #1
2290 ? :? "Taste druecken "
2300 IF PEEK(764)=255 THEN 2300
2310 POKE 764,255
2320 ?`">"
2330 GOTO 480
2340 REM  SPEICHERN
2350 REM -----------
2360 POKE 764,255
2370 POKE 54286,64:REM DLI AUS
2380 POKE 53248,0:REM P0-2 XWERTE
2390 POKE 53249,0
2400 POKE 53250,0
2410 POSITION 15,3:? "1) PRGR."
2420 POSITION 15,4:? "2) DATA"
2430 POSITION 15,5:? "3) FILE"
2440 POSITION 15,7:? "4) QUIT"
2450 A=PEEK(764)
2460 IF A=31 THEN 2510
2470 IF A=30 THEN 2740
2480 IF A=26 THEN 2810
2490 IF A=24 THEN 2660
2500 GOTO 2450
2510 REM PRGR.
2520 GOSUB 4190:REM DATEI EROEFFNEN
2530 IF PEEK(851)<>1 THEN 2660
2540 GOSUB 4420:REM ZEILENNUMMER
```

```
2550 ? #1;"0  GOSUB ";ZEILE;":REM AB HIER EIGENES PROGRAMM !"
2560 ? #1;ZEILE;" REM UNTERPRG. CHARACTERMAP ERSTELLEN"
2570 ? #1;ZEILE+1;" POKE106,PEEK(106)-4:GR.0"
2580 GOSUB 4300:REM DATA ZEILEN
2590 ? #1;ZEILE+140;" RESTORE ";ZEILE
2600 ? #1;ZEILE+141;" Z1=PEEK(106)*256"
2610 ? #1;ZEILE+142;" FOR I = Z1 TO Z1+1023"
2620 ? #1;ZEILE+143;" READD:POKEI,D"
2630 ? #1;ZEILE+144;" NEXT I"
2640 ? #1;ZEILE+145;" POKE756,PEEK(106)"
2650 ? #1;ZEILE+146;" RETURN"
2660 CLOSE #1:POKE 764,255
2670 POKE 54286,192:REM DLI AN
2680 FOR I=3 TO 7
2690 POSITION 15,I:? "        "
2700 NEXT I
2710 POKE 53249,108:REM P1 XWERT
2720 POKE 53250,108:REM P2 XWERT
2730 RETURN
2740 REM ABSPEICHERN ALS DATA
2750 GOSUB 4190:REM OEFFNE DATEI
2760 IF PEEK(851)<>1 THEN 2660
2770 GOSUB 4420:REM ZEILENNUMMER
```



```
2780 ? #1;ZEILE;" REM DATA FUER CHARACTER
MAP"
2790 GOSUB 4300:REM DATAZEILEN GENERIEREN
2800 GOTO 2660
2810 REM  FILE
2820 REM ------
2830 GOSUB 4190:REM DATEI EROEFFNEN
2840 IF PEEK(851)<>1 THEN 2660
2850 FOR I=39936 TO 39936+1023
2860 PUT #1,PEEK(I)
2870 NEXT I
2880 GOTO 2660
2890 REM  LADEN
2900 REM -------
2910 POKE 764,255
2920 POKE 54286,64:REM DLI OFF
2930 POSITION 2,10:? "
2940 POSITION 2,10:? "D:DATEINAME   ";
2950 INPUT X$
2960 TRAP 3020
2970 OPEN #1,4,0,X$
2980 IF PEEK(851)<>1 THEN 3020
2990 FOR I=CHBASNEU TO CHBASNEU+1023
3000 GET #1,A:POKE I,A
3010 NEXT I
3020 CLOSE #1
3030 POSITION 2,10:? "
3040 POKE 54286,192:REM DLI AN
3050 GOSUB 3750:REM CHR INS SICHTF.
3060 RETURN
3070 REM  BANNER
3080 REM --------
3090 POKE 764,255
3100 POKE 53248,0:REM XWERTE P(0-2)
3110 POKE 53249,0
3120 POKE 53250,0
3130 GRAPHICS 1
3140 ? #6;"    BANNERCREATOR    "
3150 POKE 756,156:REM CHBAS
3160 TRAP 3370
3170 CLOSE #1:OPEN #1,8,0,"P:"
3180 REM EINGABEN
3190 ? #6;"Geben sie den satz   ein :"
3200 ? :INPUT X$
3210 ? #6:? #6;"Welche groesse (1-8) "
3220 INPUT VER:IF VER<1 OR VER>8
THEN 3210
oder hor.(H)"
DIR$<>"H" THEN 3230
3230 ? #6:? #6;"Ausdruck... vert.(V)
3240 INPUT DIR$:IF DIR$<>"V" AND
3250 ? #6:? #6;"Linker rand (1-72)"
3260 INPUT RAND
3270 IF RAND<1 OR RAND>72 THEN 3250
3280 ? #6:? #6;"BITTE WARTEN"
3290 FOR S=1 TO LEN(X$)
3300 C=ASC(X$(S,S))
3310 INV=0
3320 IF C>128 THEN C=C-128:INV=1
3330 GOSUB 5460:REM ASCII-DISPLAYCD
3340 GOSUB 3390:REM CHR DATEN
3350 GOSUB 3520:REM DRUCKEN
3360 NEXT S
3370 POKE 764,255:CLOSE #1
3380 GOTO 480
3390 REM BUCHSTABEN DATEN
3400 REM C:BUCHSTABE
3410 FOR I=0 TO 7
3420 W(I)=PEEK(CHBASNEU+C*8+I)
3430 NEXT I
3440 FOR I=0 TO 7:FOR J=0 TO 7
3450 Q(J,I)=0
3460 NEXT J:NEXT I
3470 FOR I=0 TO 7
3480 FOR J=0 TO 7
3490 IF W(I)>=2^(7-J) THEN Q(J,I)
=1:W(I)=W(I)-2^(7-J)
3500 NEXT J:NEXT I
3510 RETURN
3520 REM BUCHSTABE DRUCKEN
3530 REM ------------------
3540 FOR I=0 TO 7
3550 ZAEHLER=RAND
3560 A$(1)=" ":A$(255)=" ":A$(2)=A$
3570 FOR J=0 TO 7
3580 IF DIR$="V" THEN Q=Q(J,I)
3590 IF DIR$="H" THEN Q=Q(I,7-J)
3600 IF Q=0 THEN 3660
3610 FOR Z=1 TO VER
3620 CHAR=42:IF INV=1 THEN CHAR=32
3630 A$(ZAEHLER)=CHR$(CHAR)
3640 ZAEHLER=ZAEHLER+1:NEXT Z
3650 GOTO 3700
3660 FOR Z=1 TO VER
3670 CHAR=32:IF INV=1 THEN CHAR=42
3680 A$(ZAEHLER)=CHR$(CHAR)
3690 ZAEHLER=ZAEHLER+1:NEXT Z
3700 NEXT J
3710 FOR Z=1 TO VER
3720 ? A$:? #1;A$
3730 NEXT Z:NEXT I
3740 RETURN
3750 REM  CHARACTER IN SICHTFELD
3760 REM -----------------------
3770 Y=21
3780 FOR I=0 TO 7
3790 BYTE=PEEK(CHBASNEU+CHRNR*8+I)
3800 FOR J=1 TO 4
3810 POKE P1BASE+Y,BYTE
3820 Y=Y+1:NEXT J
3830 NEXT I
3840 A=USR(ADR(CURS$),0,WY)
3850 RETURN
3860 REM  CHARACTER IN GRAFIKEN
3870 REM ----------------------
3880 C=CHRNR:GOSUB 5560
3890 POSITION 26,8:? "CHR$(";C;")"
3900 POSITION 35,8:? "'  '"
3910 POKE 38308,CHRNR
3920 FOR I=38357 TO 38382 STEP 5
3930 POKE I,CHRNR:POKE I+1,CHRNR
3940 POKE I+3.CHRNR
```

```
3950 NEXT I
3960 C1=CHRNR:C2=CHRNR+128
3970 FOR I=38552 TO 38591 STEP 6:POKE I,C1:POKE I+3,C2:NEXT I
3980 FOR I=38592 TO 38631 STEP 6:POKE I,C1:POKE I+3,C2:NEXT I
3990 IF C1>=64 THEN C1=0:C2=C1:C3=C1:C4=C1:GOTO 4010
4000 C2=C1+64:C3=C1+128:C4=C1+192
4010 FOR I=38632 TO 38672 STEP 12:POKE I,C1:POKE I+3,C2
4020 POKE I+6,C3:POKE I+9,C4
4030 NEXT I
4040 RETURN
4050 REM  CURSOR WAEHLEN
4060 REM ---------------
4070 WX=64:WY=62
4080 A=USR(ADR(CURS$),WX,WY)
4090 CHRNR=0
4100 S=STICK(0)
4110 IF STRIG(0)=0 THEN RETURN
4120 IF S=14 AND WY>62 THEN WY=WY-4:CHRNR=CHRNR-32:GOTO 4170
4130 IF S=13 AND WY<73 THEN WY=WY+4:CHRNR=CHRNR+32:GOTO 4170
4140 IF S=11 AND WX>64 THEN WX=WX-4:CHRNR=CHRNR-1:GOTO 4170
4150 IF S=7 AND WX<188 THEN WX=WX+4:CHRNR=CHRNR+1:GOTO 4170
```

```
4160 GOTO 4100
4170 A=USR(ADR(CURS$),WX,WY)
4180 GOTO 4100
4190 REM  DATEI EROEFFNEN
4200 REM ----------------
4210 POKE 764,255
4220 POSITION 2,10:POKE 764,255
4230 ? "                                     "
4240 POSITION 2,10:? "D:DATEINAME    ";
4250 INPUT X$
4260 TRAP 4280
4270 OPEN #1,8,0,X$
4280 POSITION 2,10:? "
4290 RETURN
4300 REM  DATA ZEILEN GENERIEREN
4310 REM ----------------------
4320 A=5
4330 FOR I=39936 TO 39936+1023 STEP 8
4340 ? #1;ZEILE+A;"DATA ";
4350 FOR J=0 TO 7
4360 ? #1;PEEK(I+J);
4370 IF J<7 THEN ? #1;",";
4380 NEXT J
4390 A=A+1:? #1
4400 NEXT I
4410 RETURN
4420 REM  ZEILENNUMMER
4430 REM -------------
4440 POSITION 2,10:? "WELCHE ANFANGSZEILE ";
4450 INPUT ZEILE
4460 POSITION 2,10:? "
4470 RETURN
4480 REM  MASCHINENPROGRAMM TRANS
4490 REM -----------------------
4500 DATA 104,169,0,133,203,133,205,169,156,133
4510 DATA 206,173,244,2,133,204,177,203,145,205
4520 DATA 230,203,208,2,230,204,230,205,208,8
4530 DATA 230,206,165,206,201,160,240,3,56,176
```

```
4540 DATA 231,96,-1
4550 RESTORE 4500:GOSUB 5640
4560 TRANS$=A$:RETURN
4570 REM  MR DREHEN
4580 REM -----------
4590 DATA 104,104,104,170,169,0,133,212,169,156
4600 DATA 133,213,224,0,240,16,24,165,212,105
4610 DATA 8,133,212,165,213,105,0,133,213,202
4620 DATA 208,240,169,128,141,163,6,162,0,160
4630 DATA 7,177,212,45,163,6,240,7,56,62
4640 DATA 192,6,56,176,4,24,62,192,6,136
4650 DATA 16,235,78,163,6,232,224,8,208,225
4660 DATA 160,7,185,192,6,145,212,136,16,248
4670 DATA 96,-1
4680 RESTORE 4590:GOSUB 5640
4690 DREHEN$=A$:RETURN
4700 REM  VBV ROUTINE
4710 REM -------------
4720 DATA 104,104,170,104,168,169,7,32,92,228
4730 DATA 96,216,206,168,6,208,22,169,22,141
4740 DATA 168,6,173,192,2,208,7,169,10,141
4750 DATA 192,2,208,5,169,0,141,192,2,173
4760 DATA 167,6,201,12,240,13,206,167,6,173
4770 DATA 167,6,141,4,212,88,76,98,228,169
4780 DATA 15,141,167,6,141,4,212,173,166,6
4790 DATA 201,40,144,5,169,0,141,166,6,141
4800 DATA 3,152,238,166,6,76,98,228,-1
4810 RESTORE 4720:GOSUB 5640
4820 VBV$=A$
4830 A=USR(ADR(VBV$),ADR(VBV$)+11)
4840 RETURN
4850 REM  MR SPIEGELN
4860 REM -------------
4870 DATA 104,104,104,170,169,0,133,212,169,156
4880 DATA 133,213,224,0,240,16,24,165,212,105
4890 DATA 8,133,212,165,213,105,0,133,213,202
4900 DATA 208,240,160,7,177,212,162,7,74,46
4910 DATA 163,6,202,16,249,173,163,6,145,212
4920 DATA 136,16,237,96,-1
4930 RESTORE 4870:GOSUB 5640
4940 SPIEGELN$=A$:RETURN
4950 REM  MR CLEAR
4960 REM ----------
4970 DATA 104,169,0,133,203,169,152,133,204,160
4980 DATA 0,169,0,145,203,230,203,208,8,230
4990 DATA 204,165,204,201,160,240,3,56,176,237
5000 DATA 96,-1
5010 RESTORE 4970:GOSUB 5640
5020 CLEAR$=A$:RETURN
5030 REM  DATA CURSOR
5040 REM -------------
5050 DATA 104,104,104,141,161,6,104,104,141,162
5060 DATA 6,24,169,0,109,160,6,133,208,169
5070 DATA 154,105,0,133,209,160,0,169,0,145
5080 DATA 208,200,192,4,208,247,24,169,0,109
5090 DATA 162,6,133,208,169,154,105,0,133,209
5100 DATA 160,0,185,176,6,145,208,200,192,4
5110 DATA 208,246,173,161,6,141,0,208,173,162
5120 DATA 6,141,160,6,96,-1
5130 RESTORE 5050:GOSUB 5640
5140 CURS$=A$:RETURN
5150 REM  INVERS MR
5160 REM -----------
5170 DATA 104,104,104,170,169,0,133,212,169,156
```

Die Bände der Programmbibliothek erscheinen getrennt für programmierbare Taschenrechner und Mikrocomputer und enthalten ausgetestete Programme zu jeweils einem ausgewählten Themenschwerpunkt, der in einem einführenden Beitrag vorgestellt wird, oder für einen aktuellen Rechner. Dabei wird der jeweilige Entwicklungsstand der Rechnertechnik berücksichtigt.
Die Programmbibliothek ist mithin als Ergänzung der Programmsammlung im Mikrocomputer-Jahrbuch anzusehen.
Die Programme sind, ausgehend von einer konkreten Aufgabenstellung, in der Regel in ihrem Ablauf beschrieben und durch ausgeführte Beispiele ergänzt. Wenn es nötig scheint, sind auch theoretische Grundlagen für die Programmierung erläutert.
Durch die graphischen, tabellarischen oder in Textform gegebenen Ablaufbeschreibungen wird die Übertragbarkeit auf andere Rechnertypen erleichtert, so daß die wirtschaftliche Nutzung der einzelnen Bände möglich ist.
An Hand gleichartiger Aufgabenstellungen wird fallweise auch die unterschiedliche Arbeitsweise verschiedener Rechnertypen aufgezeigt.





```
5180 DATA 133,213,224,0,240,16,24,165,212,105
5190 DATA 8,133,212,165,213,105,0,133,213,202
5200 DATA 208,240,160,7,177,212,73,255,145,212
5210 DATA 136,16,247,96,-1
5220 RESTORE 5170:GOSUB 5640
5230 INVERS$=A$:RETURN
5240 REM  SPLIT MR
5250 REM ----------
5260 DATA 104,104,104,141,169,6,104,104,141,170
5270 DATA 6,104,104,141,173,6,169,21,141,171
5280 DATA 6,169,128,141,172,6,172,170,6,240
5290 DATA 12,24,173,171,6,105,4,141,171,6
5300 DATA 136,208,244,174,169,6,240,6,78,172
5310 DATA 6,202,208,250,172,171,6,162,4,185
5320 DATA 128,154,77,172,6,153,128,154,200,202
5330 DATA 208,243,169,0,133,212,169,156,133,213
5340 DATA 174,173,6,240,16,24,165,212,105,8
5350 DATA 133,212,165,213,105,0,133,213,202,208
5360 DATA 240,172,170,6,177,212,77,172,6,145
5370 DATA 212,96,-1
5380 RESTORE 5260:GOSUB 5640
5390 SPLIT$=A$:RETURN
5400 REM  DLI INTERRUPT
5410 REM ---------------
5420 DATA 72,138,72,169,156,162,15,141,10,212
5430 DATA 141,9,212,142,23,208,104,170,104,64,-1
5440 RESTORE 5400:GOSUB 5640
5450 INTER$=A$:RETURN
5460 REM  UMRECHNUNG ASCII-DISPLAY
5470 REM -------------------------
5480 REM C:ASCII IN
5490 REM C:DISPLAY OUT
5500 IF C<32 THEN C=C+64:RETURN
5510 IF C<96 THEN C=C-32:RETURN
5520 IF C<128 THEN RETURN
5530 IF C<160 THEN C=C+64:RETURN
5540 IF C<224 THEN C=C-32:RETURN
5550 RETURN
5560 REM  UMRECHNUNG DISPLAY-ASCII
5570 REM -------------------------
5580 IF C<64 THEN C=C+32:RETURN
5590 IF C<96 THEN C=C-64:RETURN
5600 IF C<128 THEN RETURN
5610 IF C<192 THEN C=C+32:RETURN
5620 IF C<224 THEN C=C-64:RETURN
5630 RETURN
5640 REM  STRING ERZEUGEN
5650 REM ----------------
5660 A$(1)=CHR$(0):A$(255)=CHR$(0)
5670 A$(2)=A$
5680 ZAEHLER=1
5690 READ D:IF D=-1 THEN RETURN
5700 A$(ZAEHLER)=CHR$(D)
5710 POKE 53279,0
5720 ZAEHLER=ZAEHLER+1:GOTO 5690
5730 REM  TITEL
5740 REM -------
5750 POKE 54276,15
5760 A$=" ZEICHENEDITOR   1985   Homecomputer  ZEICHENEDITOR   1985   Homecom
puter    "
5770 ZAEHLER=1536
5780 FOR I=1 TO LEN(A$)
5790 C=ASC(A$(I,I))
5800 GOSUB 5460
5810 POKE ZAEHLER,C:ZAEHLER=ZAEHLER+1
5820 NEXT I
5830 RETURN
```

# Wer ist der Schnellste im ganzen Land?

## CPU testete die Geschwindigkeit von 28 Home- und Personalcomputern

In CPU 3/85 haben wir das Testergebnis unseres großen Benchmarktests mit 17 Rechnern veröffentlicht. In der Zwischenzeit haben wir einige neue Rechner getestet und können nun die Ergebnisse von 28 Computern abdrukken. Diesmal konnten wir auch die neuen MSX-Rechner sowie zwei weitere neue Computer berücksichtigen. Bei der Vorstellung des PC 128 und des Atari 130 XE auf der Hannover-Messe haben wir diese brandneuen Rechner den Benchmark-Programmen unterworfen. Hier sind die Testprogramme und die Ergebnisse:

---

**Benchmark I:**

```
10 FOR I=1 TO 100
20 A=SQR (I)
30 A=SIN (I)
40 A=1/I
50 NEXT I
60 PRINT „FERTIG“
70 END
```

**Benchmark II:**

```
10 FOR I=3 TO 199 STEP 2
20 FOR J=3 TO SQR (I)
   STEP 2
30 A=I/J
40 IF INT (A)=A THEN 70
50 NEXT J
60 PRINT I
70 NEXT I
80 PRINT „FERTIG“
90 END
```

| Name | Benchmark I | Benchmark II |
|---|---|---|
| Acorn BBC | 4.1 | 3.1 |
| Acorn Electron | 6.1 | 4.8 |
| Alphatronic PC | 21.5 | 19.5 |
| Apple II | 8.1 | 8.7 |
| Atari 600 | 18.4 | 17.1 |
| Atari 800 XL | 18.4 | 17.1 |
| Atari 130 XE | 19.1 | 16.8 |
| Casio FP-200 | 17.7 | 15.3 |
| Casio FP-1000 | 8.3 | 21.2 |
| CE-TEC MPC 80 MSX | 21.9 | 18.0 |
| Commodore C-16/116 | 7.9 | 9.3 |
| Commodore VC20 | 7.7 | 8.0 |
| Commodore C64 | 9.1 | 9.7 |
| Commodore PC128 | 9.8 | 11.5 |
| Commodore Plus4 | 8.0 | 9.0 |
| Fujitsu FM 7 | 3.0 | 5.5 |
| IBM PC | 4.3 | 5.3 |
| Laser 310 | 8.3 | 11.2 |
| Laser 2001 | 5.3 | 6.5 |
| Schneider CPC 464 | 4.7 | 6.5 |
| Sharp MZ 800 | 6.3 | 6.5 |
| Sinclair Spectrum | 17.6 | 19.0 |
| Sony Hit-Bit MSX | 22.4 | 17.8 |
| SVI 318/328 | 27.9 | 23.7 |
| SVI 728 MSX | 21.9 | 18.0 |
| TI 99/4A | 26.8 | 31.1 |
| VZ 200 | 7.8 | 9.0 |
| Z9001 (VEB Robotron) | 5.6 | 10.0 |

(Alle Zeitangaben in Sekunden und ohne Gewähr!)

Neuer Spitzenreiter bei Test I ist ganz klar der Fujitsu FM 7, der bereits seit 1 bis 2 Jahren auf dem Markt ist. Er besitzt eine 6809 – CPU und ist mit Hilfe einer Z-80-Erweiterung auch CP/M-fähig.

Überrascht waren wir über das schlechte Abschneiden der MSX-Rechner. Die drei getesteten Geräte lagen alle im gleichen Zeitbereich.

Der neue (kleine) Atari 130 XE, der Nachfolger des 800 XL, hat in etwa die gleiche Rechengeschwindigkeit wie sein Vorgänger. Auch der PC 128 von Commodore hat im 128er-Modus scheinbar keine größere Geschwindigkeit als der C 64, obwohl seine Ergebnisse keineswegs zu den Enttäuschungen zu zählen sind.

Auf der Hannover-Messe hatten wir auch die Möglichkeit, den Z9001 vom VEB Robotron aus der DDR zu testen. Das Ergebnis können Sie der Tabelle entnehmen. Es ist uns nicht bekannt, ob dieser Homecomputer auf dem bundesdeutschen Markt vertrieben wird.

Leider konnte der 68000-Rechner Atari ST 520 noch nicht getestet werden, da auf dem CEBIT-Stand des Herstellers keine endgültige Basic-Version vorgeführt werden konnte.

Zu den Benchmark-Tests allgemein muß noch gesagt werden, daß die Ergebnisse niemals die realen Verhältnisse fehlerfrei wiedergeben können. Die Ergebnisse einzelner Rechner hängen unter anderem auch von der Rechengenauigkeit ab. So hat der Z9001 nur eine Genauigkeit von 6 Stellen, während der Apple II auf 9 Dezimalstellen genau rechnet, was natürlich weitaus mehr Zeit in Anspruch nimmt. Benchmarks sagen auch nichts über Graphikfähigkeit und Tastatur aus, doch diese Details kann man Prospekten und „normalen“ Computertests entnehmen, während Geschwindigkeitsvergleiche auf so breiter Basis bisher (unseres Wissens) noch nicht veröffentlicht worden sind.

# CPU-Club-Vorstellung: Sharp-User-Club

Wir, das sind ca. 20 Leute mit dem MZ-700. Unsere Aktivitäten entsprechen denen der anderen Clubs. Wir treffen uns einmal im Monat zum Erfahrungsaustausch und verbreiten ein Clubinfo. Wir arbeiten mit Basic/Pascal und CP/M und suchen laufend Mitglieder vorwiegend aus dem Raum Karlsruhe.

Wir sind dem Dachverband VDC (Verband deutscher Computer-Clubs) angeschlossen und suchen auch ganze Clubs, die sich in Landesverbänden organisieren wollen.

Anschrift:
Sharp-User-Club
Dornröschenweg 15
7500 Karlsruhe 21

# Schneider: CPC-PAINTER

**Der Schneider CPC 464 ist grafisch hochtalentiert – doch was hilft's, wenn Sie nicht das richtige Programm haben, um diese künstlerischen Fähigkeiten auch voll auszunutzen? Der CPC-PAINTER schafft hier Abhilfe!**

Sobald Sie das nicht allzu lange Programm abgetippt haben und es mit RUN starten, sehen Sie eine kleine Auswahlliste. Die Nummer 3, „Beenden", scheidet als Wahlmöglichkeit vorläufig aus, denn sonst hätten Sie sich von vornherein nicht die Mühe des Abtippens gemacht. Nummer 2, „Laden", klingt so, als ob da eine Datei vorhanden sein müßte. Muß auch, ist aber nicht. Also malen Sie zuerst ein Bildchen, mit der Nummer 1.

Das Menü verschwindet, der Bildschirm wird schwarz, und am unteren Bildschirmrand erscheint ein weißer Balken. In Wirklichkeit sind das einzelne Karos – Ihre Farbtöpfe, vorerst alle mit weißer Farbe gefüllt. Tippen Sie ein T, und der erste Farbtopf beginnt zu blinken. Ein Druck auf die Rechts- oder Linkspfeil-Taste, und ein anderer Topf blinkt. Betätigen Sie eine beliebige Taste (günstig, weil greifbar, liegen die Leertaste oder die Kommataste des Zahlenblocks), und die Farbe ändert sich. Sollte der Farbtopf scheinbar verschwinden – keine Sorge: Sie haben ihn nur mit schwarzer Farbe gefüllt. Wenn Sie sich für eine Farbe entschieden haben, können Sie mit Links- oder Rechtspfeil eine weitere Farbe festsetzen oder mit COPY abschließen.

Erst mit F aber können Sie aus Ihren Farbtöpfen einen auswählen, mit dessen Farbe Sie zeichnen wollen. Mit Links- oder Rechtspfeil findet die Wahl statt, irgendeine Taste bestätigt. Wenn Sie schon einen Teil des Bildes gemalt haben, werden Sie feststellen: Die in einer bestimmten Farbe gemalten Bildteile blinken bei T und F ebenfalls mit. Das ist praktisch, denn nehmen wir an, Sie haben mit derselben Tinte eine grüne Eingangstür zu einem Haus und eine grüne Baumkrone gezeichnet. Wenn Sie nun beides blinken sehen, werden Sie merken, daß Sie sich nicht mehr für eine violette Haustür entscheiden können, indem Sie die Tinte nachträglich ändern, da sonst der Baum höchst unnatürlich aussähe.

Nach diesem Vorgeplänkel kommen wir nun endlich zum Zeichnen. L ist ein wichtiger Befehl: Mit ihm können Sie, handgesteuert, eine Linie ziehen, und zwar von der momentanen Position des Grafikcursors (das ist der blinkende Punkt, der nach dem Tippen eines Grafikbefehls auftaucht) aus. Es stehen Ihnen acht Richtungen zur Verfügung, die mit den Tasten des Ziffernblocks angesprochen werden. Die Zahlen im Bereich der 5 stellen diese Richtungen dar, mit der Taste 5 können Sie das Zeichnen der Linie abbrechen. Um exakt zeichnen zu können, haben diese Tasten **keine** Wiederholungsfunktion. Für längere Linien können Sie jedoch die entsprechenden Tasten der Schreibmaschinentastatur benutzen, wo diese Funktion existiert.

B ist ein verwandter Befehl, nur daß Sie hier Linien in – normalerweise – dreifacher Breite haben. Allerdings mit Ausnahme bestimmter Richtungen. Damit lassen sich tolle dreidimensionale Effekte erzielen. Experimentieren Sie doch damit! Gelöscht wird beim PAINTER ausschließlich durch Übermalen – und auch hier ist B von Nutzen, wenn es um größere Flächen geht.

M bedeutet „move" und bewegt lediglich den Grafikcursor an eine Stelle auf dem Bildschirm. Dazu werden die Pfeiltasten verwendet. Wenn Sie dort angelangt sind, wo Sie hinwollten, betätigen Sie COPY. Von der erreichten Stelle aus können Sie dann mit L oder B eine Linie ziehen. Alle nun folgenden Befehle beinhalten bereits die Bewegung an den Anfangspunkt; hier ist „Move" nicht nötig.

S für „Spiegeln" ist ein besonders praktischer Befehl: Sie können vollständig achsensymmetrische Gebilde erzeugen, indem Sie eine Figur an einer horizontalen oder vertikalen Achse spiegeln. Steuern Sie mit den Pfeilen an den Anfangspunkt der Achse und betätigen Sie COPY, und zwar – das ist wichtig! – zweimal. Mit einer der Pfeiltasten legen Sie sodann die Richtung der Achse fest. Betätigen Sie die Taste so lange oder so oft, bis die Achse (blinkend eingeblendet) die gewünschte Länge erreicht hat. Dann folgt wieder ein Druck auf COPY. Steuern Sie als letztes den Grafikcursor von der Achse weg (wieder mit den Pfeiltasten, versteht sich – das wird im folgenden nicht mehr extra erwähnt werden), und zwar so, daß Sie damit die Höhe (oder Breite) des Vierecks anzeigen, das gespiegelt werden soll. Ein Beispiel zur Verdeutlichung: Sie haben eine horizontale Achse, d. h. von links nach rechts. Steuern Sie den Grafikcursor unter die Achse, wobei seine horizontale Position keine Rolle spielt, so wird alles, was von dem Rechteck mit der Achse als Grundlinie und dem Abstand Cursor – Achse als Höhe eingegrenzt wird, an der Achse nach oben gespiegelt. Wenn Sie glauben, das noch nicht ganz verstanden zu haben, probieren Sie's aus.

Nehmen Sie nun einmal an, Sie möchten einen Wald aus kleinen Laub- und Nadelbäumen zeichnen. Nun ist es aber recht eintönig, zwanzig Bäume jeder Sorte, die alle gleich aussehen, auf den Bildschirm zu malen. Der Befehl U löst Ihr Problem: Damit können Sie rechteckige Bildschirmausschnitte vervielfachen. Steuern Sie nacheinander die linke untere und die rechte obere Ecke des Ausschnitts an, den Sie kopieren wollen, und betätigen Sie jeweils COPY. Fahren Sie dann mit dem Cursor an die Stelle, an der die linke untere Ecke später stehen soll. Ein weiterer Druck auf COPY, und in ein paar Sekunden steht Ihr Wald, wobei Sie nur jeweils einen einzigen Laub- und Nadelbaum zeichnen mußten!

Flächen ausmalen kann der PAINTER natürlich auch. A lautet sinnigerweise der Befehl dafür. Wichtig ist, daß die Fläche vollständig von der Farbe, in der Sie sie ausmalen wollen (d. h. der aktuellen Zeichenfarbe), umrahmt ist, sonst läuft die Farbe aus und verunstaltet evtl. das ganze Bild. Für solche und andere Situationen haben Sie jedoch eine „Notbremse": zweimal ESC, Sie hören einen kurzen Ton und können den nächsten Grafikbefehl eingeben. Jede laufende Operation wird dadurch gestoppt und der Cursor wieder nach links unten gesetzt.

Aber zurück zum Ausmalen: Eine Fläche wird von der Cursorposition aus zunächst senkrecht nach oben abgesucht. Wird dort ein Punkt der Umrandung gefunden, so geht die Suche in gleicher Weise nach unten weiter. Es ist dann natürlich ungünstig, wenn Sie z. B. von der Ecke eines Dreiecks aus dieses Dreieck auffüllen wollen, da der Computer schon nach wenigen Reihen auf die Begrenzung stoßen wird. Steuern Sie vielmehr immer die höchsten bzw. breitesten Stellen, bei Figuren mit vielen

Bögen die schmalsten an und füllen Sie von dort aus auf. Bei komplizierten Formen kann es nötig sein, daß Sie die Fläche in mehreren Schritten von verschiedenen Stellen aus ausmalen müssen.

C ist ein Befehl, den Sie weniger oft benötigen werden. Damit ist es möglich, die Farbe des Grafikcursors zu ändern, falls dieser auf einer bestimmten Hintergrundfarbe schlecht zu erkennen sein sollte. Sobald Sie C getippt haben, erscheint der Cursor links unten über den Farbtöpfen. Jede beliebige Taste ändert seine Farbe, COPY beendet. Sollte er kurz einmal verschwinden, so haben Sie gerade die Farbe Schwarz erwischt. Ein weiterer Tastendruck und Sie sehen ihn wieder blinken.

Die Taste R werden Sie als eine praktische Hilfe schätzenlernen: Damit können Sie nämlich die oft benötigte Form eines Rechtecks zeichnen. Leider können Sie es nicht rotieren lassen, um z. B. Rauten zu erzeugen, aber das ist auch nur selten vonnöten. Steuern Sie, um ein Rechteck zeichnen zu lassen, einfach die untere linke Ecke an und fixieren Sie sie mit der COPY-Taste. Wenn Sie nun mit den Pfeiltasten weitersteuern, sehen Sie die Ecke blinken. Kennzeichnen Sie auf dieselbe Weise die linke obere Ecke. Mit Betätigen von COPY erscheint auch schon das Rechteck.

Neben der Fähigkeit, Rechtecke zeichnen zu können, erlaubt Ihnen das Programm auch die automatische Darstellung von Ellipsen. O (für „Oval") ist dafür die richtige Taste. Sie müssen hier, wie bei R, einige Punkte markieren. Natürlich beginnen Sie mit dem Mittelpunkt. Danach steuern Sie nach rechts oder links und legen so die Länge einer Halbachse der Ellipse fest. Hierbei können Sie zwar nach oben oder unten steuern; beim Zeichnen wird jedoch nur die seitliche Abweichung vom Zentrum berücksichtigt. Sobald diese Achse gesetzt ist, steuern Sie nach oben oder unten (und zwar wieder vom Mittelpunkt aus; der Cursor springt automatisch zurück) – wobei Sie hier entsprechend auch links oder rechts steuern können, ohne daß dies berücksichtigt wird – und legen die Länge der zweiten Halbachse fest. Gleich nach dem Druck auf COPY sehen Sie, wie die Ellipse gezeichnet wird.

Auch n-Ecke können Sie zeichnen, allerdings nicht automatisch. Das hat den Vorteil, daß auch unregelmäßige Formen möglich werden. E heißt der benötigte Befehl. Fixieren Sie, wie gehabt, einen Eckpunkt mittels der Pfeile und COPY. Wenn Sie anschließend die Leertaste betätigen, können sie eine weitere Ecke setzen. Mit jeder anderen Taste (bewährt hat sich, der Bequemlichkeit halber, ein zweiter Druck auf COPY) werden die Eckpunkte miteinander verbunden, und zwar in der Reihenfolge, in der sie gesetzt wurden. Der letzte Punkt wird dabei automatisch mit dem ersten verbunden, damit eine geschlossene Figur entsteht. Wenn Sie die Markierungen dicht genug setzen, werden auch Wellenzüge möglich. Maximal können Sie 80 Punkte plazieren; nach dem achtzigsten wird die Figur automatisch gezeichnet. Aber so viele benötigen Sie selten.

Wahrscheinlich benötigen Sie aber nicht immer eine geschlossene Figur. Manchmal kann die Verbindung vom letzten zum ersten Punkt sogar störend wirken, weil man sie wieder ausradieren müßte. Die Taste V macht Ihnen hier das Leben leichter. Sie bedeutet „Verbinde". Die Eingabe der Punkte erfolgt wie bei E; nur die erwähnte Verbindungslinie wird nicht gezeichnet.

Was Ihnen nun noch fehlt, ist eine bequeme Möglichkeit, Strecken zu zeichnen. Da S schon belegt ist, wurde als Abkürzung G gewählt – was „Gerade" bedeuten soll, auch wenn mancher Mathematiker da protestieren mag. Praktisch ist G dasselbe, als ob Sie E anwenden würden und nur zwei Punkte setzten. Nur wird hier die Strecke gezeichnet, sobald der zweite Punkt plaziert ist, und die Betätigung der Leertaste entfällt. Also: Komfortabler und als Abkürzung leichter zu merken.

Zwei weitere Optionen runden das Ganze ab. W („Wegwerfen") ist eine davon. Ein Druck auf diese Taste löscht die Grafik und zeigt wieder die Hauptauswahl. Damit Sie aber Ihre Grafiken auch abspeichern können, wurde noch die Abkürzung K eingeführt. Logischerweise bedeutet dies „Kassette" und gestattet Ihnen, eine Grafik auf dem dadurch bezeichneten Medium zu speichern. Geben Sie einen kurzen Namen für Ihr Bild an und betätigen Sie **unbedingt** die REC- und PLAY-Taste, **bevor** Sie den Namen ENTER bestätigen. Beachten Sie auch, daß Sie beim Zeichnen noch 2–3 Punkte über den Farbtöpfen freilassen, sonst wird bei der Texteingabe der untere Rand der Zeichnung gelöscht.

Jetzt kommt auch Punkt 2 des Hauptmenüs ins Spiel. Damit können Sie Grafiken wieder laden. Auch hier unbedingt zuerst die PLAY-Taste und dann ENTER betätigen. Alle Files werden nämlich, um die Grafik nicht zu verunstalten, mit einem Ausrufungszeichen (!) vor dem Namen gespeichert, das Sie jedoch **nicht** eingeben müssen. Beim Laden bleibt der Bildschirm zunächst schwarz, bevor das Bild urplötzlich in voller Farbenpracht erscheint. Wenn Sie sodann die Taste B betätigen, können Sie das Bild weiterbearbeiten. Jede andere Taste zeigt das Hauptmenü.

Gezeichnet wird übrigens immer mit Mode 0, wobei Sie dreizehn Farben zur Verfügung haben. Die übrigen drei, über deren Verbleib Sie sich vielleicht wundern werden, benötigt das Programm für den Bildschirm, den Grafikcursor und Textanzeigen.

Einige Tips für Ihre Bilder: Am besten gehen Sie immer von einfachen Grundformen aus, auf denen Sie dann aufbauen. Zum Beispiel: Für eine Landschaft mit Bergen am Horizont setzen Sie ein grünes und ein blaues Rechteck aufeinander (R) und füllen diese in der jeweiligen Farbe aus (A). Mit E können Sie am oberen Rand des grünen Rechtecks einen Wellenzug darstellen, den Sie unten mit der Rechteckkante bündig abschließen. Das Ganze wird dann ebenfalls ausgefüllt. Sicher haben Sie aber selbst noch eine Menge Ideen.

Zum Abschluß noch eine praktische Anwendung: erzeugen Sie mit dem CPS-PAINTER Titelbilder für selbstgeschriebene, längere Programme. Speichern Sie folgendes kurzes Programm am Anfang einer Kassette.

```
10 MODE 0: FOR i=0 TO
   15: INK i, 0: NEXT i:
   BORDER 0: PAPER 0:
   CLS
20 LOAD "!(hier der Name
   des Bildes)", &C000
30 OPENIN"! (hier der
   Name des Bildes) .I": For
   i=3 TO 15: INPUT ≠9,f:
   INK i,f: NEXT i: CLO-
   SEIN
40 RUN "!(hier der Name
   Ihres Programms)"
```

Hinter diesem Programm speichern Sie mit dem PAINTER ein passendes Titelbild ab und danach schließlich Ihr Programm. Der Benutzer muß nun nur noch (bei zurückgespulter Kassette) das Band starten, wie das für die Begrüßungskassette im Handbuch beschrieben ist, und er wird automatisch von einem Titelbild begrüßt. Das Abspeichern kann stets mit SPEED WRITE 1 geschehen, da auch der PAINTER die Grafiken so ablegt.

**Kurzreferenzkarte CPC-Painter:**

**siehe S. 40**

**Hauptauswahl**
1: Zeichnen (s. dort)
2: Laden: Filenamen angeben (! zu Beginn nicht nötig); noch **vor** Druck auf ENTER die PLAY-Taste einrasten. B verzweigt in Teil „Zeichnen" (s. u.), andere Taste Hauptmenü
3: Beenden

**Zeichnen**
T: Tinte setzen. „Farbtopf" mit Linkspfeil und Rechtspfeil wählen (aktuelle Tinte blinkt); bel. Taste ändert Farbe, COPY beenden.
F: Farbe wählen. Mit Linkspfeil und Rechtspfeil Farbtopf wählen, bel. Taste akzeptiert. **Beachte:** T füllt nur die Farbtöpfe, erst F wählt die Farbe zum Zeichnen!
L: Linie ziehen. Gesteuert wird mit den Zahlentasten; der Stern um die 5 gibt die Richtung an, die 5 selbst beendet. Für Wiederholfunktion die entspr. Tasten der Schreibmaschinentastatur benutzen!
B: Breite Linie. Steuerung wie bei L; verwendet für dreidimensionale Effekte oder zum großflächigen „Radieren".
M: Move. Pfeile bewegen den Grafikcursor, COPY beendet.
S: Spiegeln. Anfangspunkt der Achse ansteuern, dann **zweimal** COPY. Richtung und Länge der Achse mit Links/Rechts- bzw. Auf/Ab-Pfeil bestimmen, mit COPY beenden. Punkt mit Pfeiltasten im rechten Winkel von der Achse wegbewegen und mittels COPY die Ausschnittbreite bzw. -höhe festlegen.
U: Übertragen. Mit Pfeilen und COPY der Reihe nach ansteuern und fixieren: linke untere und rechte obere Ecke des Quellausschnitts sowie linke untere Ecke des Zielausschnitts.
A: Ausfüllen. In eine von der Zeichenfarbe **vollständig** umrahmte Fläche steuern. COPY startet. Möglichst immer die schmalste bzw. höchste Stelle aufsuchen, evtl. Ausfüllen in mehreren Schritten.
C: Cursorfarbe. Cursor erscheint links unten. Bel. Taste ändert Farbe, COPY akzeptiert.
R: Rechteck. Linke untere und rechte obere Ecke fixieren. (Pfeile, COPY)
O: Oval. Erst Mittelpunkt setzen, dann nach rechts/links 1. Halbachse. Cursor springt wieder zum Mittelpunkt. Dann nach oben/unten 2. Halbachse setzen. Pfeile und COPY benutzen.
E: Eck. Bis zu 80 Punkte werden in der Reihenfolge des Setzens miteinander verbunden, der letzte automatisch mit dem ersten. Punkt setzen mit Pfeilen und COPY, dann Leertaste, um einen weiteren Punkt zu plazieren, oder erneut COPY (bzw. bel. andere Taste), um das n-Eck zu zeichnen.
V: Verbundener Linienzug. Wie E, nur ohne Verbindung vom letzten zum ersten Punkt.
G: Gerade, d. h. eigentlich Strecke. Anfangs- und Endpunkt mit Pfeilen und COPY setzen.
W: Wegwerfen. Löscht gesamte Grafik, kehrt zurück zum Grundmenü.
K: Kassettenspeicherung. Namen eingeben und noch vor Druck auf ENTER die REC- und PLAY-Taste betätigen. Achtung: Text benötigt unteren Bildschirmrand, dort nichts zeichnen!

„Notbremse": Unerwünschte Operationen können durch ESC-ESC abgebrochen werden.

# Variablenliste CPC-Painter

| | |
|---|---|
| ti(1..13) | Zum Zeichnen verwendete Tinten |
| eckx(1..80) | x-Koordinaten der Eckpunkte bei E und V |
| ecky(1..80) | dto., y-Koordinaten |
| ecke(1..80) | Farbe, die ursprünglich an (eckx;ecky) stand |
| i,j | Laufvariablen in Schleifen |
| e$ | als Eingabe erlaubte Tasten |
| t$ | Taste, welche betätigt wurde |
| n$ | Dateiname |
| x/y | Koordinaten des Grafikcursors |
| cf | Farbe des Cursors |
| bf | Bildschirmfarbe (immer 0) |
| tin | Index für gewählte Tinte |
| zf | INK zum Zeichnen |
| ef | urspr. Farbe an der Position des Grafikcursors |
| xs/ys | enth. x/y zur späteren Wiederverwendung |
| ef$ | durch Spiegelachse verdeckte Bildschirmfarben |
| xdir/ydir | Richtung für Spiegelachse |
| t1$ | wie t$; verwendet, wenn t$ noch benötigt wird |
| xe/ye | wie xs/ys; verwendet, wenn xs/ys noch benötigt |
| z | Zählvariable |
| xu/yu | linke untere Ecke bei U oder R |
| xo/yo | rechte obere Ecke bei U oder R |
| eu/eo | linke untere Ecke des Zielquadrats bei U |
| lx | Start einer Linie bei A |
| rx | Ende einer Linie bei A |
| eu | bei R: Farbe an der rechten unteren Ecke |
| xm/ym | Koordinaten des Mittelpunkts bei O |
| r1/r2 | Radien bei O |
| ez | Zähler für Ecken bei E oder V |
| ax/ay | Anfangspunkt der Strecke bei G |

```
100 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
110 '      CPC - PAINTER                    Version 1.2                '
120 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
130 ' (c) 1985  by    BIG BUG Software                 $               '
140 '                 Alwin Ertl                      $$$              '
150 '                 Mais 66                        $$$$$             '
160 '                 8497 Neukirchen                 $$$              '
170 '                 Tel. 09947/468                   $               '
180 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
190 '
200 'Vorbereitungen
210 DIM ti(13),eckx(80),ecky(80),ecke(80):MODE 0:BORDER 0:INK 0,0:INK 1,26:INK 2
,26,0:SPEED INK 15,15:FOR i=3 TO 15:INK i,26:ti(i-2)
=26:NEXT i:SPEED WRITE 1
220 ON BREAK GOSUB 1670
230 'Menue
```

```
240 CLS:PEN 1:PRINT TAB(4);"CPC  PAINTER":PRINT:PRINT"by BIG BUG Software":PRINT
:PRINT:PRINT:PRINT
250 PRINT"1: Zeichnen":PRINT:PRINT"2: Laden":PRINT:PRINT"3: Ende":PRINT:PRINT:PR
INT:PRINT
260 e$="123":GOSUB 1540:PRINT t$;:ON VAL(t$) GOTO 360,280,340
270 'Laden
280 FOR i=3 TO 15:INK i,0:NEXT i
290 CLS:INPUT "Name";n$:n$=UPPER$(n$):IF LEFT$(n$,1)<>"!" THEN n$="!"+n$
300 CLS:LOAD n$,&C000:OPENIN n$+".I":FOR i=1 TO 13:INPUT #9,ti(i):INK 2+i,ti(i):
NEXT i:CLOSEIN
310 t$=INKEY$:IF t$="" THEN 310
320 IF UPPER$(t$)="B" THEN 370 ELSE 240
330 'Ende
340 CLS:PRINT"Auf Wiedersehen...":PRINT:PRINT:PRINT:END
350 'Zeichnen
360 CLS
370 x=0:y=10:cf=26:bf=0:tin=1:zf=3
380 FOR i=3 TO 15:FOR j=-19+(i-2)*20 TO (i-2)*20:MOVE j,0:DRAW j,8,i:NEXT j,i
390 SOUND 1,200,10,14:e$="TFLBMSUACRDEVGWK":GOSUB 1540:ON INSTR(e$,t$) GOTO 410,
480,540,650,760,780,990,1040,1140,1190,1230,1300,138
0,1450,1480,1500
400 'Tinte setzen
410 INK 2+tin,ti(tin),bf
420 t$=INKEY$:IF t$="" THEN 420 ELSE INK 2+tin,ti(tin)
430 IF t$=CHR$(224) THEN 390
440 IF t$=CHR$(243) THEN tin=tin+1:IF tin>13 THEN tin=1:GOTO 410 ELSE 410
450 IF t$=CHR$(242) THEN tin=tin-1:IF tin<1 THEN tin=13:GOTO 410 ELSE 410
460 ti(tin)=ti(tin)+1:ti(tin)=(ti(tin) MOD 27):GOTO 410
470 'Farbe waehlen
480 INK zf,ti(zf-2),bf
490 t$=INKEY$:IF t$="" THEN 490 ELSE INK zf,ti(zf-2)
500 IF t$=CHR$(243) THEN zf=zf+1:IF zf>15 THEN zf=3:GOTO 480 ELSE 480
510 IF t$=CHR$(242) THEN zf=zf-1:IF zf<3 THEN zf=15:GOTO 480 ELSE 480
520 INK zf,ti(zf-2):GOTO 390
530 'Linie ziehen
540 ef=TEST(x,y):PLOT x,y,2:e$="123456789":GOSUB 1540:PLOT x,y,zf:ON VAL(t$) GOT
O 550,560,570,580,390,590,600,610,620
550 x=x-4:y=y-2:GOTO 630
560 y=y-2:GOTO 630
570 x=x+4:y=y-2:GOTO 630
580 x=x-4:GOTO 630
590 x=x+4:GOTO 630
600 x=x-4:y=y+2:GOTO 630
610 y=y+2:GOTO 630
620 x=x+4:y=y+2
630 DRAW x,y,zf:GOTO 540
640 'Breite Linie
650 ef=TEST(x,y):PLOT x,y,2:e$="123456789":GOSUB 1540:PLOT x,y,ef:xs=x:ys=y:ON V
AL(t$) GOTO 660,670,680,690,390,700,710,720,730
660 x=x-4:y=y-2:GOTO 740
670 y=y-2:GOTO 740
680 x=x+4:y=y-2:GOTO 740
690 x=x-4:GOTO 740
700 x=x+4:GOTO 740
710 x=x-4:y=y+2:GOTO 740
720 y=y+2:GOTO 740
730 x=x+4:y=y+2
740 DRAW x,y,zf:PLOT xs+4,ys+2,zf:DRAW x+4,y+2,zf:PLOT xs-4,ys-2,zf:DRAW x-4,y-2
,zf:GOTO 650
750 'Move
760 GOSUB 1570:IF t$<>CHR$(224) THEN 760 ELSE 390
770 'Spiegeln
780 GOSUB 1570:IF t$<>CHR$(224) THEN 780
790 GOSUB 1570:ef$="":ef$=ef$+CHR$(ef):xs=x:ys=y:PLOT x,y,2
800 ef$="":ef$=ef$+CHR$(ef):xs=x:ys=y:PLOT x,y,2
```

```
810 t$=INKEY$:IF t$="" THEN 810 ELSE xdir=0:ydir=0
820 IF t$=CHR$(243) THEN xdir=4
830 IF t$=CHR$(242) THEN xdir=-4
840 IF t$=CHR$(241) THEN ydir=-2
850 IF t$=CHR$(240) THEN ydir=2
860 t1$=INKEY$
870 WHILE (t1$=t$ OR t1$="") AND LEN(ef$)<255
880 IF t1$="" THEN t1$=INKEY$:GOTO 900
890   x=x+xdir:y=y+ydir:ef$=ef$+CHR$(TEST(x,y)):PLOT x,y,2:t1$=INKEY$
900 WEND
910 xe=x:ye=y
920 GOSUB 1570:IF t$<>CHR$(224) THEN 920
930 IF xdir THEN 960
940 FOR i=ys TO ye STEP SGN(ye-ys)*2:FOR j=x TO xs STEP SGN(xs-x)*4:PLOT xs+(j-x
+1),i,TEST(xs-(j-x+1),i):NEXT j,i
950 z=0:FOR i=ys TO ye STEP SGN(ye-ys)*2:z=z+1:PLOT xs,i,ASC(MID$(ef$,z,1)):NEXT
 i:GOTO 390
960 FOR i=xs TO xe STEP SGN(xe-xs)*4:FOR j=y TO ys STEP SGN(ys-y)*2:PLOT i,ys+(j
-y+1),TEST(i,ys-(j-y+1)):NEXT j,i
970 z=0:FOR i=xs TO xe STEP SGN(xe-xs)*4:z=z+1:PLOT i,ys,ASC(MID$(ef$,z,1)):NEXT
 i:GOTO 390
980 'Uebertragen
990 GOSUB 1570:IF t$<>CHR$(224) THEN 990 ELSE xu=x:yu=y:eu=TEST(x,y):PLOT x,y,2
1000 GOSUB 1570:IF t$<>CHR$(224) THEN 1000 ELSE xo=x:yo=y:eo=TEST(x,y):PLOT x,y,
2
1010 GOSUB 1570:IF t$<>CHR$(224) THEN 1010 ELSE PLOT xu,yu,eu:PLOT xo,yo,eo
1020 FOR i=xu TO xo STEP 4:FOR j=yu TO yo STEP 2:PLOT x+(i-xu),j+(y-yu),TEST(i,j
):NEXT j,i:GOTO 390
1030 'Ausfuellen
1040 GOSUB 1570:IF t$<>CHR$(224) THEN 1040 ELSE sx=x:sy=y
1050 y=y+2:IF TEST(x,y)=zf OR y>400 THEN y=sy+2:GOTO 1090
1060 x=x-4:IF TEST(x,y)<>zf AND x>=0 THEN 1060 ELSE lx=x:x=sx
1070 x=x+4:IF TEST(x,y)<>zf AND x<=640 THEN 1070 ELSE rx=x:x=sx
1080 PLOT lx,y,zf:DRAW rx,y,zf:GOTO 1050
1090 y=y-2:IF TEST(x,y)=zf OR y<0 THEN 390
1100 x=x-4:IF TEST(x,y)<>zf AND x>=0 THEN 1100 ELSE lx=x:x=sx
1110 x=x+4:IF TEST(x,y)<>zf AND x<=640 THEN 1110 ELSE rx=x:x=sx
1120 PLOT lx,y,zf:DRAW rx,y,zf:GOTO 1090
1130 'Cursorfarbe
1140 ef=TEST(0,10):PLOT 0,10,2
1150 t$=INKEY$:IF t$="" THEN 1150
1160 IF t$=CHR$(224) THEN PLOT 0,10,ef:GOTO 390
1170 cf=cf+1:cf=(cf MOD 26)+1:INK 2,cf,0:GOTO 1150
1180 'Rechteck
1190 GOSUB 1570:IF t$<>CHR$(224) THEN 1190 ELSE xu=x:yu=y:eu=TEST(x,y):PLOT x,y,
2
1200 GOSUB 1570:IF t$<>CHR$(224) THEN 1200 ELSE xo=x:yo=y:PLOT xu,yu,eu
1210 PLOT xu,yu,zf:DRAW xo,yu,zf:DRAW xo,yo,zf:DRAW xu,yo,zf:DRAW xu,yu,zf:GOTO
390
1220 'Oval
1230 GOSUB 1570:IF t$<>CHR$(224) THEN 1230 ELSE xm=x:ym=y:em=TEST(x,y):PLOT x,y,
2
1240 GOSUB 1570:IF t$<>CHR$(224) THEN 1240 ELSE xr=x:yr=y:er=TEST(x,y):PLOT x,y,
2:IF x>xm THEN r1=x-xm ELSE r1=xm-x
1250 x=xm:y=ym
1260 GOSUB 1570:IF t$<>CHR$(224) THEN 1260 ELSE IF y>ym THEN r2=y-ym ELSE r2=ym-
y
1270 PLOT xm,ym,em:PLOT xr,yr,er
1280 DEG:PLOT xm+r1,ym,zf:FOR i=0 TO 360:DRAW xm+COS(i)*r1,ym+SIN(i)*r2,zf:NEXT
i:GOTO 390
1290 'Eck
1300 ez=0
1310 ez=ez+1:IF ez>80 THEN ez=80:GOTO 1350
1320 GOSUB 1570:IF t$<>CHR$(224) THEN 1320 ELSE eckx(ez)=x:ecky(ez)=y:ecke(ez)=T
EST(x,y):PLOT x,y,2
```

```
1330 t$=INKEY$:IF t$="" THEN 1330
1340 IF t$=" " THEN 1310
1350 FOR i=1 TO ez:PLOT eckx(i),ecky(i),ecke(i):NEXT i
1360 PLOT eckx(1),ecky(1),zf:FOR i=2 TO ez:DRAW eckx(i),ecky(i),zf:NEXT i:DRAW e
ckx(1),ecky(1),zf:GOTO 390
1370 'Verbundener Linienzug
1380 ez=0
1390 ez=ez+1:IF ez>80 THEN ez=80:GOTO 1430
1400 GOSUB 1570:IF t$<>CHR$(224) THEN 1400 ELSE eckx(ez)=x:ecky(ez)=y:PLOT x,y,2
1410 t$=INKEY$:IF t$="" THEN 1410
1420 IF t$=" " THEN 1390
1430 PLOT eckx(1),ecky(1),zf:FOR i=2 TO ez:DRAW eckx(i),ecky(i),zf:NEXT i:GOTO 3
90
1440 'Gerade
1450 GOSUB 1570:IF t$<>CHR$(224) THEN 1450 ELSE ax=x:ay=y:PLOT x,y,2
1460 GOSUB 1570:IF t$<>CHR$(224) THEN 1460 ELSE PLOT ax,ay,zf:DRAW x,y,zf:GOTO 3
90
1470 'Weg damit
1480 CLEAR:RUN
1490 'Kassette
1500 LOCATE 1,25:INPUT "Name";n$:IF LEFT$(n$,1)<>"!" THEN n$="!"+n$
1510 FOR i=0 TO 14:MOVE 0,i:DRAW 640,i,0:NEXT i
1520 n$=UPPER$(n$):SAVE n$,b,&C000,(&FFFF-&C000):OPENOUT n$+".I":FOR i=1 TO 13:P
RINT#9,ti(i):NEXT i:CLOSEOUT:GOTO 380
1530 'UP Eingabe
1540 t$=INKEY$:IF t$="" THEN 1540
1550 t$=UPPER$(t$):IF INSTR(e$,t$)=0 THEN 1540 ELSE RETURN
1560 'UP Grafikcursor steuern
1570 ef=TEST(x,y):PLOT x,y,2
1580 t$=INKEY$:IF t$="" THEN 1580 ELSE PLOT x,y,ef
1590 IF t$=CHR$(243) THEN x=x+4:GOTO 1630
1600 IF t$=CHR$(242) THEN x=x-4:GOTO 1630
1610 IF t$=CHR$(241) THEN y=y-2:GOTO 1630
1620 IF t$=CHR$(240) THEN y=y+2:GOTO 1630 ELSE RETURN
1630 IF x>639 THEN x=0 ELSE IF x<0 THEN x=640
1640 IF y>399 THEN y=0 ELSE IF y<0 THEN y=400
1650 GOTO 1570
1660 'Abbruch
1670 x=0:y=10:GOTO 390
```

# Tips und Tricks CPC-464: Merge-Befehl bei der Verwendung der Diskettenstation DDI-1

Wer schon einmal mit einem „Merge"-Befehl versucht hat, Programme miteinander zu verbinden, wird mit der Diskettenstation DDI-1 schlechte Erfahrungen gemacht haben. Im Diskettenbetriebssystem der Floppy hat sich bei der Entwicklung ein Fehler eingeschlichen. Fast jedesmal, wenn ein Programm nachgeladen werden soll, meldet sich der Rechner mit der Fehlermeldung „EOF met".

Dies liegt daran, daß das nachzuladende Programm vom Diskettenbetriebssystem, nach dem Kennzeichen EOF (Ende der Datei $EO), untersucht wird. Da leider auch die Zeilennummer, Länge und Pointer mituntersucht werden, wird sehr oft diese Kennzeichnung gefunden. Wir können dies beheben, indem wir Programme als ASCII-Code auf die Disketten schreiben.

Beispiel: SAVE „PROGRAMM", A

Das Laden dieser Programme unterscheidet sich nicht von den normal abgespeicherten Programmen. Der einzige Nachteil besteht darin, daß im ASCII-Code abgespeicherte Programme mehr Speicherplatz auf der Diskette belegen.

# ZX-Spectrum: Mit Maschinensprache zum Bildschirm

Teil 2

Im letzten Heft haben wir gelernt, wie man mit Hilfe von Rom-Routinen Zeichen auf dem Bildschirm darstellen kann. Wer das kurze Beispielprogramm eingegeben hat, wird bemerkt haben, daß die Ausgabe immer in der augenblicklich aktiven Bildschirmfarbe erfolgte. Wie lassen sich nun die Farben beeinflussen? Zum Ausdruck verwandten wir die RST 10 (10 hex 16 dez) Routine. Der Code des auszugebenden Zeichens wurde in das Register A geladen. Wir können aber auch die Steuer-Codes ausgeben, gefolgt von der Farbnummer.

Die Farbsteuercodes sind:

| Dez | Hex | Bedeutung |
|---|---|---|
| 16 | 10 | INK |
| 17 | 11 | PAPER |
| 18 | 12 | FLASH |
| 19 | 13 | BRIGHT |
| 20 | 14 | INVERSE |
| 21 | 15 | OVER |

Nehmen wir also das Beispiel aus dem letzten Heft

```
3E 02      LD A 2       Bildschirmkanal setzen
CD 01 16   CALL 1601
06 18      LD B 18      PRINT Position setzen
0E 21      LD C 21
CD D9 0D   CALL 0DD9
3E 10      LD A 10      Steuerzeichen für INK
D7         RST 10       Ausgeben
3E 02      LD A 2       Farbe Rot
D7         RST 10       Ausgeben
3E 41      LD A 41      Code für A
D7         RST 10       Ausgeben
C9         RET          Zurück zu BASIC
```

Nach Ablauf des Programmes steht der Buchstabe A in Rot an Position 0,0. Experimentieren Sie ruhig mit dem Farbwert und verwenden Sie auch andere Steuerzeichen. Sie werden sehen, wie einfach der Umgang mit dem Bildschirm ist. Sie können auch weitere RST 10-Befehle an das Programm anhängen, die neuen Zeichen erscheinen jeweils rechts vom letzten Zeichen; genauso, als hätten Sie in einem PRINT-Kommando ein Semikolon eingefügt.

Noch zwei nützliche Adressen im Umgang mit dem Bildschirm:

```
CALL 0D6B   Löscht den Bildschirm
CALL 0DFE   Scrollt den Bildschirm nach oben
```

Bei beiden Routinen brauchen keine Daten angegeben werden. Vom BASIC aus können Sie mit:

```
RANDOMIZE USR 3435
RANDOMIZE USR 3582
```

gestartet werden.

Kommen wir nun zur Graphik. Punkte können Punkte auf den Bildschirm plotten, indem Sie eine Routine bei 22E5 Hex 8933 DEZ verwenden. Vor Anwendung der Routine müssen die Register B und C mit den X,Y-Koordinaten geladen werden.

```
Register B   Y-Koordinate (0-175 dez) (0-AF hex)
Register C   X-Koordinate (0-225 dez) (0-FF hex)
```

Um einen Punkt in der Mitte des Bildschirms zu setzen, können wir folgendes Programm benutzen.

```
06 55      LD B 55      Y-Koordinate
0E 80      LD C 80      X-Koordinate
CD E5 22   CALL 22E5    Punkt plotten
C9         RET
```

Um Punkte zu löschen, bieten sich zwei Möglichkeiten an: Erstens: Sie verwenden OVER 1. Das kann durch Einfügen des Befehls: SET 0 (IY+87) erreicht werden. Zweitens: Sie verwenden INVERSE 1. Der Befehl hierfür lautet: SET 2 (IY+87). Beide Befehle können jederzeit aufgehoben werden, indem man die entsprechenden BITs zurücksetzt; also RES 0 (IY+87) und RES 2 (IY+87).

---

# SMP-T830-Al-Siemens-Mini-Terminal

Folien-Tastatur und LCD-Anzeige sind die äußeren Kennzeichen eines Mini-Terminals, das Siemens für Mikrocomputersysteme vorstellt. Die neue Kleinstation (SMP-T830-Al) ist kaum größer oder schwerer als ein Lesebuch, dennoch steht der Leistungsumfang dem eines herkömmlichen Terminals mit Bildschirm und Tisch-Tastatur kaum nach. CMOS-Bausteine sind die inneren Merkmale des Mini-Terminals: Mikrocomputer 80C31, EPROM (bis 8 Kbyte), RAM (bis 16 Kbyte). Mit dem Gerät können Programmteile aus dem Entwicklungslabor an den Einsatzort gebracht oder dort geändert werden. Erleichtert wird auch die Arbeit des Kundendienstes (Inbetriebnahme, Wartung).

Das kleine Terminal arbeitet mit einer Batterie netzunabhängig bis zu acht Stunden. Eine zweite Batterie puffert das CMOS-RAM. Beim Ausschalten des Gerätes bleiben erhalten: Terminalkonfiguration (Baudrate etc.), Festlegung des Zeichensatzes, Tastaturbelegung und Makros. Eine V.24-Schnittstelle verbindet das Terminal mit der Entwicklungsstation bzw. mit dem Mikrocomputersystem (SMP, AMS, SYS9000 o. a.). Für längere Strecken kann die Schnittstelle auch als 20-mA-Stromschleife benutzt werden.

Die LCD-Anzeige besteht aus vier Zeilen zu je 40 Zeichen (5 x 7 Punktmatrix), die fensterartig den kompletten Speicherinhalt vorführen. Zwischen deutschen und ASCII-Zeichensatz kann umge-

schaltet werden. Die alphanumerische Folien-Tastatur (wasserdicht!) ermöglicht den Zugang zur menuegesteuerten Benutzeroberfläche mit acht Betriebsarten von „online“ (Terminal-Betrieb) bis zu „send“ (Aussenden eines erstellten Textes). Die Übertragungsgeschwindigkeit kann von 75 Baud bis zu 96000 Baud variiert werden.




# TI – „PRO-LITE“ professionelle Computerleistung aus dem Aktenkoffer

Texas Instruments stellt einen wirklich tragbaren, mit allen Funktionen ausgerüsteten PC vor, der in einem Aktenkoffer untergebracht werden kann und zudem noch Raum für Papier läßt.

Mit diesem neuen Mitglied der PC-Familie unterstreicht Texas Instruments einmal mehr sein Engagement auf dem Markt der Professional Computer. Sämtliche Geräte (Tischgeräte, Transportable und Pro-Lite) sind untereinander kompatibel.

Die Portabilität ist grundlegende Bedingung für Geschäftsleute, die oft unterwegs sind.

Vor 15 Jahren hat TI als erstes Unternehmen mit der SILENT 700 (TM)-Produktfamilie ein tragbares Computer- Terminal vorgestellt.

Der TI Pro-Lite stellt eine natürliche Verbindung der tragbaren Terminals und Professional Computer Produkte TI's dar.

Pro-Lite ist bei einer Größe von 75 x 290 x 330 mm 4,8 kg leicht. Durch den flachen Flüssigkristall-Bildschirm im Standardformat mit 25 Zeilen à 80 Zeichen wird Pro-Lite zum wirklich tragbaren Gerät. Die hohe Auflösung des Bildschirms macht die Darstellung von Grafiken möglich.

Als Option bietet TI einen externen Farb- oder Monochrom-Video-Bildschirm an. Durch seine Kompatibilität mit dem TIPC ermöglicht Pro-Lite den Einsatz von über 600 der häufigsten „Third Party“-Anwendungen des MS (TM)-DOS Betriebssystems. Mit Hilfe des Emulators 931 kann Pro-Lite als intelligentes Terminal an TI Bürocomputer angeschlossen werden und über DFÜ-Protokolle auch mit allen Großrechnern kommunizieren.

Pro-Lite verfügt über einen RAM-Arbeitsspeicher von 256 KB, erweiterbar auf 768 KB. Für Programm- und Datenspeicherung ist Pro-Lite mit einem 3,5-Zoll-Disketten-Laufwerk mit einer Speicherkapazität von 720 KB ausgestattet. Damit lassen sich auch komplexe Lösungen wie beispielsweise die integrierte Software Open Access (TM) von „SPI“ (Software Products International) einsetzen. Pro-Lite Anwendern kommt auch die Pionierstellung TI's im Bereich der künstlichen Intelligenz zugute.

Auf „Personal Consultant“ erstellte Expertensysteme, die bereits auf dem TIPC verfügbar sind, können nunmehr wirklich mobil eingesetzt werden.

Die Zentraleinheit des Computers ist ein 16-Bit-Mikroprozessor vom Typ 80C88, der auf Wunsch durch den numerischen Coprozessor 8087 ergänzt werden kann. Mit Hilfe eines Verbindungskabels kann Pro-Lite Daten-Files mit den anderen Mitgliedern der TI-Familie sowie des IBM PCs austauschen.

Das modulare Design von Pro-Lite ermöglicht, daß der Anwender auf einfache Weise ein System zusammenstellen kann, das seinen individuellen Erfordernissen entspricht. Eine ansteckbare Erweiterungsbox erlaubt den Anschluß eines zweiten Diskettenlaufwerks sowie eines Akkus. Darüber hinaus stehen zwei weitere Steckplätze für entweder eine RS 232 sync/async Schnittstelle, eine Schnittstelle für einen externen Farb- oder Monochrom-Bildschirm oder ein programmierbares Software-Modul für individuelle Anwendungen zur Verfügung.

Die Basiseinheit enthält einen dem Industriestandard entsprechenden Parallel Port für den Anschluß eines Druckers. In der tragbaren Form bietet TI einen 45-Zeichen-Thermodrucker, der durch das Hauptgerät betrieben wird und auf Normalpapier oder Thermopapier mit 40, 80 oder 132 Zeichen/Zeile drucken kann. Pro-Lite ist mit allen Matrix-Druckern der Serie OMNI 800 kompatibel.

Die Tastatur des Pro-Lite bietet neben 12 frei programmierbaren Funktionstasten insgesamt 79 Tasten in der üblichen Schreibmaschinennorm. Durch das zusätzliche numerische Tastenfeld werden Rechenvorgänge anwenderfreundlicher.

Mit einem RAM-Speicher von 256 KB und einem 3,5-Zoll-Disketten-Laufwerk mit 720 KB liegt der Listenpreis von Pro-Lite bei ca. DM 11 000,– exkl. MwSt.

## Ergänzung zum Anwenderprogramm „Multicolor“ für den TI-99

**Die fehlende Zahl in Zeile 910 lautet 039.**

# Hallo, liebe Rätselfreunde!

Diesmal hat sich die CPU-Redaktion was ganz Besonderes ausgedacht. Ihre Aufgabe besteht darin, die Funktion des unten abgedruckten Unterprogrammes herauszufinden.

Eine Hilfe:
Vor dem Aufruf muß eine bestimmte Variable mit einem Wert oder einem String belegt werden. Nach Ablauf des Programmes enthält eine andere Variable einen entsprechenden Ausgangswert.
Haben Sie die Funktion des Programmes herausgefunden, so notieren Sie diese sowie Ein- und Ausgangsvariable auf eine Postkarte.
Vergessen Sie bitte nicht Absender und Angabe Ihres Computersystems!

**1. Preis**
10 Kassetten aus unserem Software-Service

**2. Preis**
3 Kassetten aus unserem Software-Service

**3. Preis**
2 Kassetten aus unserem Software-Service

**4.–30. Preis**
1 Kassette aus unserem Software-Service

Tronic Verlag GmbH
Postfach 1
3444 Wehretal 1
Kennwort CPU-Happy

```
10 REM *** WAS MACHT DIESES UNTERPROGRAMM ***
20 REM
30 REM EINGANGSVARIABLE = ?
40 REM AUSGANGSVARIABLE = ?
50 REM
100 L=LEN (DU$)
110 DE=0
120 FOR I = 1 TO L
130 B$=MID$(DU$,I,1)
140 IF B$="1" THEN DE=DE+2↑(L-I)
150 NEXT I

READY.
```

*Alle richtigen Lösungen nehmen an der Verlosung der Programmkassetten teil. Mitmachen darf jeder, mit Ausnahme der Mitarbeiter des Tronic-Verlages sowie deren Angehörige. Einsendeschluß ist der 15. 6. 1985.*

*Der Rechtsweg bleibt wie üblich ausgeschlossen.*

# Deutscher Zeichensatz für C-64/MPS 802

Dieses Programm ermöglicht das Ausdrucken von „ÄÖÜäöüß" auf dem Commodore-Drucker VC 1526/MPS 802. Dies geschieht durch Einbau des Programms in die Ausgaberoutine. Immer, wenn bestimmte Codes über die Sekundäradressen 0, 1 oder 7 der Geräteadresse 4 ausgegeben werden, stoppt das Programm die Ausgabe, ändert das benutzerdefinierbare Zeichen und gibt es aus, dann wird normal weitergedruckt. Für die Definition ist es leider erforderlich, daß ein CHR$(141) ausgegeben wird. Dadurch werden alle Sonderfunktionen wie Invers- oder Breitschrift, aber auch der Gänsefüßchenmodus, abgeschaltet. Groß- oder Kleinschrift werden auch abgeschaltet; wenn man Großschrift benutzen möchte, muß man die Ausgabe über Sekundäradresse 0 laufen lassen. Benötigt man Kleinschrift, erfolgt die Ausgabe über Sekundäradresse 7. Dadurch ist das Programm vor allem für die Ausgabe von Texten, wie zum Beispiel Briefen, geeignet. Für Listings sollte man die Erweiterung wieder abschalten. Dies geschieht durch Run-Stop/Restore; vor jeder Ausgabe sollte die Erweiterung durch Sys 51712 neu initialisiert werden. Übrigens stört die Erweiterung die Ausgabe an die Floppy, Datasette oder andere Geräte nicht, sie kann von daher also durchaus eingeschaltet bleiben.

Die CHR$-Codes der Umlaute sind nicht festgelegt, jeder kann sie nach eigenen Wünschen ändern, denn die Standardzeichen, deren Codes verwendet werden, können natürlich nicht mehr ausgedruckt werden. Sie können also Zeichen, die Sie gut über die Tastatur erreichen und die Sie sonst nicht oft verwenden, in die Codetabelle eintragen. So läßt sie sich beispielsweise an Textprogramme oder selbsterstellte Zeichensätze anpassen. Die Codes im Programm stehen in der Reihenfolge von „ÄÖÜäöüß"; die Werte im Listing entsprechen den Codes bei Epson- und kompatiblen Druckern. Wenn Sie sie ändern, müssen Sie beachten, daß die Werte der Codes nicht zwischen 0–31 und nicht zwischen 128–159 liegen dürfen, da in diesen Bereichen die Steuercodes für den Drucker untergebracht sind.

Wenn Sie das Programm eingetippt haben und das erste Mal ausprobieren, werden Sie feststellen, daß der Druckkopf bei jedem Umlaut einen „Schlag" macht, durch die Druckwegoptimierung des VC 1526/MPS 802 wird daraus manchmal ein richtiges Hin- und Herlaufen des Kopfes. Wenn Sie übrigens Breitschrift einstellen, wird ein Sonderzeichen mitten im schon gedruckten Text erscheinen.

Weil das Programm die zu druckenden Zeichen zählt und da Breitschrift nicht extra berücksichtigt wird, kommt es dann zu einem „Verzählen" des Programms. Da das Programm sich das zuletzt benutzte Sonderzeichen merkt, wird es bei seiner Wiederholung nicht neu definiert, deswegen sollte man das Programm vor jedem Ausdruck auch mit Sys 51712 neu starten, da so das letzte Zeichen gelöscht wird, sonst würde, wenn der Drucker zwischendurch abgeschaltet wurde, statt des Sonderzeichens eventuell ein Leerzeichen ausgedruckt.

Trotzdem ist das Programm für die Ausgabe von Texten nahezu ideal, wenn man auf Breitschrift und Inversschrift verzichtet.

```
1000 REM DEUTSCHER ZEICHENSATZ  MPS 802
1010 REM     COPYRIGHT                BY
1020 REM          KLAUS SCHNEIDER
1030 REM           BAHNHOFSTR. 5
1040 REM            493 DETMOLD
1050 REM
1060 REM ******* POKEN DER CODES ******
1070 REM
1080 FOR I=1 TO 7
1090   READ W
1100   POKE 52043+I,W
1110 NEXT I
1120 REM
1130 REM ****** POKEN DER DATEN *******
1140 REM
1150 PRINT CHR$(147)
1160 PRINT TAB(14)"BITTE WARTEN"
1170 PRINT
1180 FOR I=51712 TO 52043
1190   READ W
1200   CO=CO+W
1210   POKE I,W
1220 NEXT I
1230 IF CO<>41096 THEN PRINT "ERROR IN DATAS"
1240 PRINT "ALLES OK."
1250 REM
1260 REM ***** STARTEN DER ROUTINE ****
1270 REM
1280 SYS 51712
1290 END
1300 REM
1310 REM **** HIER CODES EINTRAGEN ****
1320 REM
1330 DATA 091,092,093,123,124,125,126
1340 REM
1350 REM ********* DIE DATEN **********
1360 REM
1370 DATA 169,022,141,038,003,169,202
1380 DATA 141,039,003,141,083,203,162
1390 DATA 000,142,085,203,142,084,203
1400 DATA 096,072,165,154,201,004,240
1410 DATA 004,104,076,202,241,104,133
1420 DATA 097,165,185,201,255,240,017
1430 DATA 201,096,240,013,201,097,240
1440 DATA 009,201,103,240,005,165,097
1450 DATA 076,202,241,133,098,138,072
1460 DATA 152,072,165,097,041,127,201
1470 DATA 032,176,003,076,245,202,165
1480 DATA 185,201,103,208,010,173,084
1490 DATA 203,208,005,169,017,032,202
1500 DATA 241,165,097,162,006,221,076
1510 DATA 203,240,006,202,016,248,076
1520 DATA 217,202,138,205,083,203,208
1530 DATA 005,169,254,076,231,202,141
1540 DATA 083,203,165,184,072,169,141
1550 DATA 032,202,241,032,204,255,162
1560 DATA 003,189,016,203,149,183,202
```

```
1570 DATA 016,248,032,192,255,162,099
1580 DATA 032,201,255,173,083,203,010
1590 DATA 010,010,170,160,008,189,020
1600 DATA 203,032,202,241,232,136,208
1610 DATA 246,169,013,032,202,241,032
1620 DATA 204,255,169,099,032,195,255
1630 DATA 104,170,032,201,255,174,084
1640 DATA 203,240,008,169,032,032,202
1650 DATA 241,202,208,248,166,098,134
1660 DATA 185,169,254,224,103,208,021
1670 DATA 032,202,241,169,017,208,014
1680 DATA 201,034,208,010,169,255,077
1690 DATA 085,203,141,085,203,169,034
1700 DATA 238,084,203,133,097,104,168
1710 DATA 104,170,165,097,076,202,241
1720 DATA 041,127,201,013,208,012,162
1730 DATA 000,142,085,203,142,084,203
1740 DATA 165,097,208,227,165,097,044
1750 DATA 085,203,048,217,016,218,000
1760 DATA 099,005,004,158,040,072,072
1770 DATA 040,158,000,000,188,066,066
1780 DATA 066,066,188,000,000,188,002
1790 DATA 002,002,002,188,000,000,132
1800 DATA 042,042,042,156,002,000,000
1810 DATA 156,034,034,034,156,000,000
1820 DATA 000,188,002,002,004,190,000
1830 DATA 000,000,127,128,146,082,044
1840 DATA 000,000,000
```

# DEL 64 für den Commodore 64

Dieses Programm dient dazu, Programmzeilen blockweise zu löschen. Es wird normal geladen und mit RUN gestartet. Da es sich jedoch selbst löscht, sollte man es unbedingt vor dem ersten Lauf absaven.

Nach dem Programmstart wird man nach der Anfangsadresse gefragt, ab der die Maschinenroutine im Speicher abgelegt werden soll. Wenn der Bereich von $C000-$D000 nicht anderweitig benutzt wird, kann man diesen wählen (49152). Nun wird das Maschinenprogramm in den entsprechenden Speicherbereich gepoked, wobei eine Prüfsumme gebildet wird. Stimmt diese, so wird eine entsprechende Meldung ausgegeben und der Basic-Lader löscht sich selbst.

Die Routine steht nun frei zur Verfügung. Sie wird mit SYSA,Z1,Z2 aufgerufen, wobei A die vorher eingegebene Anfangsadresse ist. Z1 und Z2 stehen für die erste und die letzte zu löschende Programmzeile. Hierbei ist zu beachten, daß diese beiden Zeilen ebenfalls gelöscht werden, wenn sie vorhanden sind.

Für Z1 und Z2 können auch Zeilennummern eingegeben werden, die im Programm nicht vorkommen. Dann werden einfach die dazwischenliegenden Zeilen gelöscht. Wenn Z2 kleiner oder gleich Z1 ist, wird einfach gar nichts gelöscht.

Nach Beendigung des Löschvorganges springt die Routine ins Basic zurück.

```
100 REM  ***  DEL 64  V3  ***
110 REM  (C)1985 BY RALPH LAESSIG
120 :
130 INPUT"STARTADRESSE";A
140 FORI=ATOA+105
150 READQ:POKEI,Q:S=S+Q:NEXT
160 IFS=14245THENPRINT"ALLES OK !!":NEW
170 PRINT"FEHLER IN DATAS !!":END
180 :
190 DATA032,253,174,032,138,173,032
200 DATA247,183,072,152,072,032,253
210 DATA174,032,138,173,032,247,183
220 DATA132,253,133,254,104,133,020
230 DATA104,133,021,032,019,166,165
240 DATA095,133,251,165,096,133,252
250 DATA177,095,208,001,096,165,253
260 DATA133,020,165,254,133,021,160
270 DATA001,032,029,166,144,012,160
280 DATA000,177,095,170,200,177,095
290 DATA133,096,134,095,056,165,046
300 DATA229,096,170,232,160,000,177
310 DATA095,145,251,200,208,249,230
320 DATA252,230,096,202,208,242,032
330 DATA089,166,032,051,165,076,116
340 DATA164
```

```
2
110:    C000                    .OPT P1,00
120:    C000                    *=   $C000
                           ;
                           ;
                           ;DEL 64  V3
                           ;BLOCKWEISES LOESCHEN VON
                           ;PROGRAMMZEILEN
                           ;10.04.1985  BY RALPH LAESSIG
                           ;
                           ;
200:    C000            CHKCOM   =     $AEFD
210:    C000            FRMNUM   =     $AD8A
220:    C000            FACINT   =     $B7F7
230:    C000            ZEIG1    =     251
240:    C000            ZEIG2    =     253
                           ;
                           ;
                           ;PARAMETER HOLEN
```

```
280:    C000 20 FD AE             JSR   CHKCOM
290:    C003 20 8A AD             JSR   FRMNUM
300:    C006 20 F7 B7             JSR   FACINT   ;STARTADRESSE
310:    C009 48                   PHA
320:    C00A 98                   TYA
320:    C00B 48                   PHA            ;SICHERN
330:    C00C 20 FD AE             JSR   CHKCOM
340:    C00F 20 8A AD             JSR   FRMNUM
350:    C012 20 F7 B7             JSR   FACINT   ;ENDADRESSE
360:    C015 84 FD                STY   ZEIG2
370:    C017 85 FE                STA   ZEIG2+1  ;SICHERN

                     BEREICHSANFANG SUCHEN
400:    C019 68                   PLA
410:    C01A 85 14                STA   $14
420:    C01C 68                   PLA            ;ZEILENNUMMER SETZEN
430:    C01D 85 15                STA   $15
440:    C01F 20 13 A6             JSR   $A613    ;ZEILENADRESSE SUCHEN
450:    C022 A5 5F                LDA   $5F
460:    C024 85 FB                STA   ZEIG1    ;UND SICHERN
470:    C026 A5 60                LDA   $60
480:    C028 85 FC                STA   ZEIG1+1
490:    C02A B1 5F                LDA   ($5F),Y
500:    C02C D0 01                BNE   BE       ;KEIN PROGRAMMENDE
510:    C02E 60                   RTS

                     BEREICHSENDE SUCHEN
540:    C02F A5 FD         BE     LDA   ZEIG2
550:    C031 85 14                STA   $14
560:    C033 A5 FE                LDA   ZEIG2+1  ;ZEILENNUMMER SETZEN
570:    C035 85 15                STA   $15
580:    C037 A0 01                LDY   #$01
590:    C039 20 1D A6             JSR   $A61D    ;ADRESSE SUCHEN
600:    C03C 90 0C                BCC   W1       ;ZEILE NICHT VORHANDEN

620:    C03E A0 00                LDY   #0
630:    C040 B1 5F                LDA   ($5F),Y
640:    C042 AA                   TAX            ;ZEILE VORHANDEN
650:    C043 C8                   INY            ;=> ADRESSE VON NAECHSTER ZEILE
660:    C044 B1 5F                LDA   ($5F),Y  ;BERECHNEN
670:    C046 85 60                STA   $60
680:    C048 86 5F                STX   $5F
                          ;
                          ;LOESCHEN
710:    C04A 38            W1     SEC
720:    C04B A5 2E                LDA   46       ;LAENGE DES ZU VERSCHIEBENDEN
730:    C04D E5 60                SBC   $60      ;TEILS BERECHNEN
740:    C04F AA                   TAX
750:    C050 E8                   INX
760:    C051 A0 00                LDY   #0
770:    C053 B1 5F         L1     LDA   ($5F),Y
780:    C055 91 FB                STA   (ZEIG1),Y
790:    C057 C8                   INY
800:    C058 D0 F9                BNE   L1       ;VERSCHIEBESCHLEIFE
810:    C05A E6 FC                INC   ZEIG1+1
820:    C05C E6 60                INC   $60
830:    C05E CA                   DEX
840:    C05F D0 F2                BNE   L1
850:    C061 20 59 A6             JSR   $A659    ;CLR
860:    C064 20 33 A5             JSR   $A533    ;PRGMZEILEN NEU BINDEN
870:    C067 4C 74 A4             JMP   $A474    ;EINGABESCHLEIFE
```

# High-Resolution Graphik

## Ein Unterprogramm zum Plotten von einzelnen Punkten für den TI-99

**Erläuterungen:**

| Zeile: | Erläuterungen: |
|---|---|
| 40- 90 | Programmkopf |
| 100-110 | Erklärung |
| 120-170 | Dateneingabe |
| 180-210 | YMAX, YMIN-Berechnung |
| 220-250 | Übergabewerte XO/YO -errechnen |
| 260-270 | Tastaturabfrage<br>E = ENDE<br>W = Wiederholt Plot<br>N = Neue Werte |
| 280-320 | Berechnung der Punkte im 192 mal 256 Raster |
| 330-340 | Hexadezimalumrechnung |

Alle „!" können ersatzlos gestrichen werden.

Das Unterprogramm Call Plot kann natürlich auch anderweitig verwendet werden, z. B. zu einem Zeichenprogramm, das auch beiliegt, daß ich aber nicht näher erklären will, da es sich um exakt die Zeilen 280-340 und einer Joystickabfrage handelt. Es kommen also noch 3 Zeilen hinzu. Soll ein Punkt gesetzt werden, braucht man nur den Feuerknopf drücken. Soll der Bildschirm gelöscht werden, „A" drücken.

Das Unterprogramm kann auch noch in viele andere Grafikprogramme eingebunden werden (Zeile 280-340) und braucht dann nur mit CALL PLOT (X,Y-,)„123456789ABCDEF") aufgerufen zu werden. X für die X-Achse (1-256); Y für die Y-Achse (1-192)!!!

**Programmbeschreibung:**

Nach dem Start des Programms erscheint eine kurze Erklärung. Nach richtigem Start müssen dann der Anfangswert, Endwert, Schrittweite, YMAX, YMIN eingegeben werden. Bei der Sin-Funktion sind das z. B. Anfangswert=O, Endwert=2x-PI=6.28, Schrittweite=.05 (Normal zwischen .1 und .01), YMAX=1, YMIN=-1.

Wenn man die YMAX- und YMIN-Werte nicht im Kopf hat, dann gibt man einfach für beides „O" ein und der Rechner errechnet sich die Werte selbst. Das dauert etwas länger. Das Programm ist aber trotz allem das schnellste Plot-Programm für den TI in Basic.

Das Programm ist gegen falsche Werte weitgehend gesichert, und so kann man mit den YMAX- und YMIN-Werten etwas spielen, und man erhält tolle Verzerrungen. Werden bestimmte Grenzwerte überschritten, wie z. B. der Zeichen-Vorrat oder X- und Y-Werte, so stürzt das Programm nicht ab, sondern macht mit dem nächsten Wert weiter. Der Plot ist dann erstellt, wenn ein akustisches Signal ertönt. Wahlmöglichkeiten sind dann, siehe oben.

**Variablenliste:**

| | |
|---|---|
| YMAX | Maximum von Y |
| YMIN | Minimum von Y |
| Y1 | Y-Effektiv |
| X,Y | X,Y-Werte |
| XO,YO | Übergabewerte an SUB PLOT |
| K1,S1 | Keyrückgabevariablen |
| XE | X-ENDE |
| XA | X-ANFANG |
| S | Schrittweite |
| H§,C§ | Stringausdrücke für die CHAR-Definition |
| Q,K | Variable für Zeichen (ASCII), das gesetzt werden soll |
| V,W | Annahmevariablen des SUB PLOT |
| VG,WG,VR,WR,VO,WO | Variablen zur Errechnung der Punkte bzw. der CHARs |

```
40 !HIGH RES. PLOT                 .   2.4 Kbyt
50 !RAINER GAWRIKOW
60 !SOMMERHAUSEN 15
70 !5203 MUCH
80 !TEL.:02245/3983
90 !
100 DISPLAY AT(8,1)ERASE ALL:"HOCHAUFLOESENDER PLOT!":"_____________________":
: :"VON RAINER GAWRIKOW 2/85'": : :"!!DIE FUNKTION MUSS IN ZEILE": : :"130 DEFIN
IERT WERDEN. Z.B.:"
110 DISPLAY AT(21,1):"Y=SIN(X),DANN RUN 130!!" :: STOP
120 !EINGABE DER WERTE
130 DEF Y=SIN(X)
140 CALL SCREEN(3):: DISPLAY AT(12,1)ERASE ALL:"ANFANGSWERT ?": :"ENDWERT      ?"
: :"SCHRITTWEITE?": :"YMAX         ?": :"YMIN         ?"
150 ACCEPT AT(12,15)BEEP:XA :: ACCEPT AT(14,15)BEEP:XE :: IF XE<=XA THEN 150
160 ACCEPT AT(16,15)BEEP:S :: IF S<=0 THEN 160
170 ACCEPT AT(18,15)BEEP:YMAX :: ACCEPT AT(20,15)BEEP:YMIN :: IF YMIN>YMAX THEN
170
180 !ERRECHNEN DER GROESSTEN UND KLEINSTEN Y-WERTE
190 ON BREAK NEXT :: ON WARNING NEXT :: IF YMAX=0 AND YMIN=0 THEN 200 ELSE 210
200 FOR X=XA TO XE STEP S :: YMAX=MAX(YMAX,Y):: YMIN=MIN(YMIN,Y):: NEXT X
210 Y1=YMAX-YMIN :: IF Y1=1 THEN Y1=2
220 !ROUTINE ZUM ERRECHNEN DER WERTE IM(192*256) RASTER
230 CALL CLEAR :: FOR X=XA TO XE STEP S
240 X0=((-SGN(XA)*ABS(XA))+X)*(250/(XE-XA)):: Y0=180-ABS(YMIN*(170/Y1))-Y*(170/Y
1):: IF Y0<8 OR Y0>182 THEN 250 ELSE CALL PLOT(Y0,X0,"123456789ABCDEF")
250 NEXT X :: CALL SOUND(-500,500,10)
```

```
260 CALL KEY(0,K,S1):: IF S1=0 THEN 260 ELSE IF K=87 THEN CALL CLEAR :: GOTO 230
 ELSE IF K=78 THEN CALL CLEAR :: CALL CHARSET :: GOTO 140 ELSE IF K=69 THEN CALL
 CLEAR :: CALL CHARSET :: GOTO 100
270 CALL CLEAR :: END
280 !ROUTINE ZUR EINZELPUNKTANSTEUERUNG (192/256)
290 SUB PLOT(V,W,H$)
300 W=INT(W):: WG=INT(W/8):: WR=W-WG*8+1 :: V=INT(V):: VG=INT(V/8):: VR=V-VG*8+1
 :: W0=INT((WR/4.1-INT(WR/4.1))*4.1+.5):: V0=VR*2+INT(WR/4.1)-1
310 CALL GCHAR(VG+1,WG+1,Q):: IF Q=32 THEN K=K+1 :: Q=K+32 :: C$=RPT$("0",16)ELS
E CALL CHARPAT(Q,C$)
320 IF Q>=143 THEN SUBEXIT
330 !HEXADEZIMALUMRECHNUG
340 C$=SEG$(C$,1,V0-1)&SEG$(H$,POS(H$,SEG$(C$,V0,1),1)OR 2^(4-W0),1)&SEG$(C$,V0+
1,15):: CALL CHAR(Q,C$):: CALL HCHAR(VG+1,WG+1,Q):: SUBEND
```

# Das Floppybuch zum Apple II

**Data-Becker-Verlag, Düsseldorf; Autor: Manfred Czerwinski; ISBN-3-89011-072-X**

Die kleinen biegsamen Kunststoffscheiben – Disketten, Floppydisks oder kurz Floppies genannt – sind aus der Welt der Mikrocomputer nicht mehr wegzudenken. Das Floppy-Buch zum Apple II befaßt sich intensiv mit den Einsatzmöglichkeiten dieses Speichermediums. Der Schwerpunkt liegt bei der Behandlung der verschiedenen Dateiformen.

Der Darstellung liegt das Diskettenbetriebssystem DOS 3.3 zugrunde. Im letzten Kapitel wird das neue System ProDOS behandelt; die erweiterten Möglichkeiten und Unterschiede zu DOS werden aufgezeigt. Alle Programme, die Sie in diesem Buch finden, sind unter beiden Betriebssystemen lauffähig.

Sie sollten bereits über solide Grundkenntnisse in BASIC verfügen und mit Applesoft-Befehlen vertraut sein. Die Prinzipien der Programmierung wie beispielsweise das Formulieren von Schleifen und Verzweigungen, die Funktion von Ein- und Ausgabebefehlen oder der Umgang mit Feldern und Zeichenketten werden vorausgesetzt.

Auf dem Gebiet der Dateiverarbeitung benötigen Sie keinerlei Kenntnisse, da wir sozusagen bei „Null" anfangen. Aber auch für Fortgeschrittene, die mit sequentiellen und Random-Dateien schon gearbeitet haben, bieten insbesondere die Kapitel 5 bis 9 neue Möglichkeiten, Alternativen sowie Tips und Tricks.

Sie finden in diesem Buch eine ganze Reihe von Programmen, die Sie direkt in Ihren Apple-Computer eintippen und anwenden können. Wenn Sie beispielsweise eine Kunden-, Artikel- oder Vereinskartei benötigen, können Sie das entsprechende Programm unmittelbar einsetzen. Das Ziel des Buches liegt jedoch **nicht** darin, fertige Programme zum Abtippen zu präsentieren. Vielmehr sollen Sie die **Prinzipien** der Dateiverarbeitung kennenund verstehen lernen und den **Entwurf der Programme** Schritt für Schritt nachvollziehen können. Diese Vorgehensweise ist zwar für Sie zunächst etwas mühsamer, Sie profitieren aber wesentlich mehr davon: Sie können nämlich die Programme so modifizieren, daß diese für Ihre persönlichen Anwendungen optimal sind. Und Sie sind in der Lage, eigene Programme zu schreiben, welche auf den hier dargestellten Prinzipien beruhen. Aus diesem Grund enthält das Buch zahlreiche Abbildungen zur Verdeutlichung wichtiger Programmiermethoden; ferner sind alle Programme ausführlich dokumentiert. Das Buch ist daher Lehrbuch und Programmsammlung in einem.

**Einige Kapitel aus dem Inhaltsverzeichnis:**



# 3½ Zoll Micro-Floppy-Disk zum Mitnehmen!

## Für Olivetti M10 und Tandy TRS80-Model 100.

Daß man seinen Computer mit auf die Reise nehmen kann, ist vielen schon eine Selbstverständlichkeit. Aber wenn schon mobiler Computer – dann auch mobiler Speicher. In der Größe sollte natürlich auch alles zusammenpassen.

Damit wären wir auch schon bei der PFD-400. Der netzunabhängigen 3½ Zoll Micro-Floppy für Ihren M10 bzw. Model 100. Klein und handlich mit erstaunlicher Leistung: 3½ Zoll Drive mit Schnittstelle zu Ihrem Computer. Wiederaufladbare Ni-Cad-Batterien sowie Netzanschluß für ein Ladegerät.

Da ist Platz für jede Menge Daten – ca. 360 KB können Sie pro Diskette (formatiert) abspeichern. Die Zugriffszeit: Ganze 6 Millisekunden (Track to Track).

Die PFD-400 schließen Sie ganz einfach an Ihren Computer über das mitgelieferte Kabel an (BUS-Verbindung). Übrigens – Ihre Schnittstellen für Drucker und Modem bleiben frei!

Last but not least ist Ihre Floppy mit einem integrierten Videointerface ausgerüstet. Problemlos schließen Sie hier einen Monitor (80 x 25) oder ein TV-Gerät (40 x 25) an.

# „Hello" – Anwenderprogramm für den Apple II

Ist das Programm erst einmal **vollständig** abgeschrieben und als erstes File auf Ihrer Diskette zu finden, müssen Sie sich nach dem Einschalten des Rechners erst einmal zwischen der RETURN-Taste und der SPACE-Taste entscheiden. Drücken Sie die RETURN-Taste, befinden Sie sich im Menü. Haben Sie sich aber für die SPACE-Taste entschieden, erleben Sie eine kleine HIRES-Überraschung, die zum schnelleren Arbeiten auch weggelassen werden kann (einfach Zeilen 10–600 nicht mit abschreiben).

Dieses DOS hat von mir eine persönliche Note bekommen. Ist das bei Anwendern nicht erwünscht – einfach die Zeilen 35–290 weglassen (oder in mühseliger Kleinstarbeit das Programm umschreiben, auf daß der eigene Name erscheine).

**Zum Menü:**

Das Menü ist sehr übersichtlich. Links erscheint das Inhaltsverzeichnis der Diskette, das durch Drücken der SPACE-Taste erneut geladen werden kann.

Ist der CATALOG einmal länger als 18 Zeilen, muß man erst eine Taste drücken, um an die zur Verfügung stehenden DOS-Anweisungen (auf dem Bildschirm rechts) zu kommen. Diese sind knapp und deutlich gehalten; aus Platzgründen, und um den Anwender nicht zu verwirren.

**Vorgehensweise:**

1. **Zum Initialisierungsvorgang (3. INIT):**
   In diesem Programmteil wird zuerst nach der Diskettenkennzahl gefragt. Danach wird SLOTNR. 6 vorgegeben. Ist diese zu beanstanden, so ist diese Frage mit „N" zu beantworten und kann somit berichtigt werden. Als nächstes wird auf die Laufwerknummer verwiesen. Gleiches Verfahren wie bei der Slotnummer. Es folgt ein Hinweis, daß dieses Programm das erste auf der zu formatierenden Diskette sein wird. Als zusätzliche Absicherung wird noch einmal gefragt, ob auch die richtige Diskette im Laufwerk Nr. soundso ist.
2. **Zum Umbenennen von Programmnamen (7. RENAME):**
   Es ist nicht mit dem Komma zu verfahren, sondern wie folgt: Name des Programms, welches man umbenennen will – (RETURN) – neuer Name – (RETURN).
3. **Zur 9. SAVE**
   Dieser Teil des Programms bringt nur dann einen Vorteil, wenn man öfter einmal das erste Programm der Diskette wechseln bzw. erneuern will.

Das Ganze hat zwei große Vorteile gegenüber dem Formatieren:

1. Es wird nicht die ganze Diskette gelöscht.
2. Große Zeitersparnis.

Z. B. habe ich dauernd an dem Programm getüftelt und es immer wieder verbessert und war es leid, immer zu tippen: UNLOCK HELLO, dann evtl. DELETE HELLO – etwa wenn das Programm kürzer geworden ist –, dann SAVE HELLO usw. ...

Der Text, der dann folgt, ist etwa so zu übersetzen: „Wenn du den SAVE-Teil dieses Programms weiter verfolgst, wird das erste Programm deiner Diskette – HELLO – zerstört werden, ob es nun durch LOCK gesichert ist oder nicht. Anstatt dessen wird sich dieses Programm selbst abspeichern und automatisch sichern. Wenn du fertig bist, tippe „Y" ..."

Dieses Programm steigt auch nicht gleich aus, sollte etwa ein Programm mal nicht zu finden sein oder ähnliches. Wird einmal das Inhaltsverzeichnis durch SPACE abgefragt und ist keine Diskette im Laufwerk, läuft das Programm so lange, bis eine Diskette eingelegt oder die RESET-Taste gedrückt wird. (Vorsicht! Nicht gut für das Laufwerk!)

```
10  HGR : HGR2 : HOME : GET E$
15  IF E$ =  CHR$ (32) THEN 20
17  IF E$ =  CHR$ (13) THEN 605
18  GOTO 10
20  HCOLOR= 5: FOR I = 191 TO 1 STEP  - 3: HPLOT 279,I TO I,0: NEXT I
30  FOR I = 0 TO 191 STEP 3: HPLOT 0,I TO 88 + I,191: NEXT I
35  REM ***   R   ***
40  HCOLOR= 2: FOR I = 1 TO 6: HPLOT 45,80 TO 55 + I,90: NEXT I
50  HCOLOR= 3: FOR I = 1 TO 4: HPLOT 45,85 TO 61,65 - I: NEXT I
60  HCOLOR= 2: FOR I = 1 TO 4: HPLOT 45,60 TO 61,60 + I: NEXT I
70  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 44 + I,60 TO 45,90: NEXT I
75  REM ***   O   ***
80  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 81,60 TO 82 - I,90: NEXT I
90  HCOLOR= 2: FOR I = 1 TO 4: HPLOT 65,60 TO 81,60 + I: NEXT I
100  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 64 + I,60 TO 65,90: NEXT I
110  HCOLOR= 2: FOR I = 4 TO 1 STEP  - 1: HPLOT 80,90 TO 65,90 - I: NEXT I

120  REM ***   N   ***
130  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 101,60 TO 102 - I,90: NEXT I
140  HCOLOR= 2: FOR I = 1 TO 6: HPLOT 89,60 TO 95 + I,90: NEXT I
150  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 84 + I,60 TO 85,90: NEXT I
155  REM ***   A   ***
```

```
160  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 121,60 TO 122 - I,90: NEXT I
170  HCOLOR= 2: FOR I = 1 TO 4: HPLOT 105,60 TO 121,60 + I: NEXT I
180  FOR I = 4 TO 1 STEP  - 1: HPLOT 121,75 TO 105,75 - I: NEXT I
190  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 104 + I,60 TO 105,90: NEXT I
195  REM ***   L   ***
200  HCOLOR= 2: FOR I = 1 TO 4: HPLOT 125,90 TO 141,85 + I: NEXT I
210  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 124 + I,60 TO 125,90: NEXT I
215  REM ***   D   ***
230  HCOLOR= 2: FOR I = 1 TO 4: HPLOT 150,60 TO 161,65 + I: NEXT I
235  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 161,65 TO 162 - I,85: NEXT I
240  HCOLOR= 2: FOR I = 4 TO 1 STEP  - 1: HPLOT 161,85 TO 145,90 - I: NEXT
      I
250  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 144 + I,60 TO 145,90: NEXT I
255  REM ***       ***
260  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 173 + I,60 TO 168,70: NEXT I
265  REM ***   S   ***
270  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 190 + I,75 TO 180,90: NEXT I
280  HCOLOR= 2: FOR I = 4 TO 1 STEP  - 1: HPLOT 196,75 TO 181,75 - I: NEXT
      I
290  HCOLOR= 3: FOR I = 1 TO 6: HPLOT 190 + I,60 TO 180,75: NEXT I
295  REM ***   D   ***
300  HCOLOR= 6: FOR I = 105 TO 110: HPLOT I,95 TO I,145: NEXT
310  FOR I = 130 TO 135: HPLOT I,100 TO I,140: NEXT
320  FOR I = 100 TO 105: HPLOT 105,I - 5 TO 135,I: NEXT
330  FOR I = 135 TO 140: HPLOT 105,I + 5 TO 135,I: NEXT
335  REM ***   O   ***
340  FOR I = 141 TO 146: HPLOT I,110 TO I,160: NEXT
350  FOR I = 166 TO 171: HPLOT I,110 TO I,160: NEXT
360  FOR I = 110 TO 115: HPLOT 141,I TO 171,I: NEXT
370  FOR I = 155 TO 160: HPLOT 141,I TO 171,I: NEXT
375  REM ***   S   ***
380  FOR I = 177 TO 182: HPLOT I,125 TO I,152: NEXT
390  FOR I = 202 TO 207: HPLOT I,147 TO I,175: NEXT
400  FOR I = 125 TO 130: HPLOT 177,I TO 207,I: NEXT
410  FOR I = 170 TO 175: HPLOT 177,I TO 207,I: NEXT
420  FOR I = 147 TO 152: HPLOT 177,I TO 207,I: NEXT
430  FOR I = 1 TO 2000: NEXT
435  REM ***   ISK   ***
440  HCOLOR= 3: HPLOT 137,95 TO 143,95: HPLOT 137,108 TO 143,108: HPLOT 14
      0,95 TO 140,108
450  HPLOT 151,95 TO 145,95 TO 145,102 TO 151,102 TO 151,108 TO 145,108
460  HPLOT 153,95 TO 153,108: HPLOT 159,95 TO 153,102 TO 159,108
465  REM ***   PERATING   ***
470  HPLOT 173,123 TO 173,110 TO 179,110 TO 179,117 TO 173,117
480  HPLOT 187,110 TO 181,110 TO 181,123 TO 187,123: HPLOT 181,117 TO 184,
      117
490  HPLOT 189,123 TO 189,110 TO 195,110 TO 195,117 TO 189,117 TO 195,123
500  HPLOT 197,123 TO 197,117 TO 203,117 TO 203,123: HPLOT 198,116 TO 198,
      110 TO 202,110 TO 202,116
510  HPLOT 205,110 TO 211,110: HPLOT 208,110 TO 208,123
520  HPLOT 213,110 TO 219,110: HPLOT 216,110 TO 216,123: HPLOT 213,123 TO
      219,123
530  HPLOT 221,123 TO 221,110 TO 227,123 TO 227,110
540  HPLOT 235,113 TO 235,110 TO 229,110 TO 229,123 TO 232,123 TO 235,120:
       HPLOT 231,117 TO 235,117 TO 235,123
545  REM ***   YSTEM   ***
550  HPLOT 209,125 TO 209,132 TO 215,132 TO 215,125: HPLOT 212,132 TO 212,
      138
560  HPLOT 223,125 TO 217,125 TO 217,132 TO 223,132 TO 223,138 TO 217,138
```

```
570  HPLOT 225,125 TO 231,125: HPLOT 228,125 TO 228,138
580  HPLOT 239,125 TO 233,125 TO 233,138 TO 239,138: HPLOT 233,132 TO 236,
     132
590  HPLOT 241,138 TO 241,125 TO 244,132 TO 247,125 TO 247,138
600  FOR I = 1 TO 2000: NEXT
605  REM ***   ANFANG   ***
607  ONERR  GOTO 9000
610  HOME : TEXT : VTAB 1
615  INVERSE : FOR E = 1 TO 120: PRINT " ";: NEXT
620  VTAB 2: PRINT "          DOS - BY RONALD KREILINGER": NORMAL
640  POKE 34,3
650  PRINT  CHR$ (13) +  CHR$ (4);"CATALOG"
660  VTAB 5: HTAB 32: INVERSE : PRINT "1": NORMAL : HTAB 34: VTAB 5: PRINT
     "RUN"
670  VTAB 7: HTAB 32: INVERSE : PRINT "2": NORMAL : HTAB 34: VTAB 7: PRINT
     "LOAD"
680  VTAB 9: HTAB 32: INVERSE : PRINT "3": NORMAL : HTAB 34: VTAB 9: PRINT
     "INIT"
690  VTAB 11: HTAB 32: INVERSE : PRINT "4": NORMAL : HTAB 34: VTAB 11: PRINT
     "LOCK"
700  VTAB 13: HTAB 32: INVERSE : PRINT "5": NORMAL : HTAB 34: VTAB 13: PRINT
     "UNLOCK"
710  VTAB 15: HTAB 32: INVERSE : PRINT "6": NORMAL : HTAB 34: VTAB 15: PRINT
     "DELETE"
720  VTAB 17: HTAB 32: INVERSE : PRINT "7": NORMAL : HTAB 34: VTAB 17: PRINT
     "RENAME"
730  VTAB 19: HTAB 32: INVERSE : PRINT "8": NORMAL : HTAB 34: VTAB 19: PRINT
     "EXIT"
732  VTAB 21: HTAB 32: INVERSE : PRINT "9": NORMAL : HTAB 34: VTAB 21: PRINT
     "SAVE"
735  POKE 33,31
750  IF  PEEK ( - 16384) < 128 THEN 750
755  GET WN$
757  IF WN$ =  CHR$ (32) THEN  POKE 33,40: HOME : GOTO 650
760  IF WN$ = "1" THEN 1000
770  IF WN$ = "2" THEN 2000
780  IF WN$ = "3" THEN 3000
790  IF WN$ = "4" THEN 4000
800  IF WN$ = "5" THEN 5000
810  IF WN$ = "6" THEN 6000
820  IF WN$ = "7" THEN 7000
830  IF WN$ = "8" THEN 8000
835  IF WN$ = "9" THEN 10000
840  IF WN$ < "1" OR WN$ > "8" THEN 750
1000  REM ***   RUN   ***
1010  POKE 34,0: POKE 33,40: HOME
1020  HTAB 19: INVERSE : PRINT "RUN"
1030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21:
      PRINT "= MENU"
1070  PRINT : PRINT : INPUT "WHAT FILE? ";F$
1080 D$ =  CHR$ (4)
1090  PRINT D$;"RUN";F$
1100  GOTO 1000
2000  REM ***   LOAD   ***
2010  POKE 34,0: POKE 33,40: HOME
2020  HTAB 18: INVERSE : PRINT "LOAD"
2030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21:
      PRINT "= MENU"
2070  PRINT : PRINT : INPUT "WHAT FILE? ";F$
2080 D$ =  CHR$ (4)
```

```
2090  PRINT D$;"LOAD";F$
2100  GOTO 2000
3000  REM ***   INIT   ***
3010  POKE 34,0: POKE 33,40: HOME
3020  HTAB 18: INVERSE : PRINT "INIT"
3030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21:
      PRINT "= MENU"
3065 S = 6:D = 1
3070  VTAB 8: INPUT "VOLUME?  ";V
3090  HTAB 1: VTAB 12: PRINT "SLOT # 6 - RIGHT?": VTAB 12: HTAB 20
3091  GET E$
3092  IF E$ = "N" THEN 3095
3093  GOTO 3110
3095  VTAB 12: HTAB 28: INPUT "SLOT # ";S
3110  HTAB 1: VTAB 16: PRINT "DRIVE # 1 - RIGHT?": VTAB 16: HTAB 20
3112  GET E$
3113  IF E$ = "N" THEN 3117
3115  GOTO 3120
3117  VTAB 16: HTAB 27: INPUT "DRIVE # ";D
3120  VTAB 19: HTAB 1: PRINT "THIS PROGRAM WILL BE THE 1ST ON THE DISKYOU'
     RE FORMATTIN'.": PRINT
3122  VTAB 22: HTAB 1: PRINT "DO YOU HAVE THE RIGHT DISK IN DRIVE "D"?": VTAB
     22: HTAB 39
3124  GET E$: IF E$ = "N" THEN 3124
3600  PRINT  CHR$ (13) +  CHR$ (4)"INITHELLO,V";V;",S";S;",D";D
3700  PRINT  CHR$ (13) +  CHR$ (4)"LOCK HELLO"
3800  GOTO 3000
4000  REM ***   LOCK   ***
4010  POKE 34,0: POKE 33,40: HOME
4020  HTAB 18: INVERSE : PRINT "LOCK"
4030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21:
      PRINT "= MENU"
4070  PRINT : PRINT : INPUT "WHAT FILE? ";F$
4080 D$ =  CHR$ (4)
4090  PRINT D$;"LOCK";F$
4100  GOTO 4000
5000  REM ***   UNLOCK   ***
5010  POKE 34,0: POKE 33,40: HOME
5020  HTAB 17: INVERSE : PRINT "UNLOCK"
5030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21:
      PRINT "= MENU"
5070  PRINT : PRINT : INPUT "WHAT FILE? ";F$
5080 D$ =  CHR$ (4)
5090  PRINT D$;"UNLOCK";F$
5100  GOTO 5000
6000  REM ***   DELETE   ***
6010  POKE 34,0: POKE 33,40: HOME
6020  HTAB 17: INVERSE : PRINT "DELETE"
6030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21:
      PRINT "= MENU"
6070  PRINT : PRINT : INPUT "WHAT FILE? ";F$
6080 D$ =  CHR$ (4)
6090  PRINT D$;"DELETE";F$
6100  GOTO 6000
7000  REM ***   RENAME   ***
7010  POKE 34,0: POKE 33,40: HOME
7020  HTAB 17: INVERSE : PRINT "RENAME"
7030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21:
      PRINT "= MENU"
```

```
7070  PRINT : PRINT : INPUT "WHAT FILE? ";O$
7080  PRINT : PRINT : INPUT "NEW NAME?  ";N$
7100 D$ =  CHR$ (4)
7110  PRINT D$;"RENAME ";O$;",";N$
7120  GOTO 7000
8000  REM ***   EXIT   ***
8010  POKE 34,0
8020  POKE 33,40
8100  HOME
8999  END
9000  INVERSE :D$ =  CHR$ (7)
9001  INVERSE :D$ =  CHR$ (7)
9002  IF  PEEK (222) = 1 THEN  PRINT : PRINT : HTAB 10: PRINT D$;"LANGUAGE
      NOT AVAILABLE": GOSUB 9090
9010  IF  PEEK (222) = 4 THEN  PRINT : PRINT : HTAB 14: PRINT D$;"WRITE PR
     OTECTED": GOSUB 9090
9020  IF  PEEK (222) = 6 THEN  PRINT : PRINT : HTAB 14: PRINT D$;"FILE NOT
      FOUND": GOSUB 9090
9030  IF  PEEK (222) = 8 THEN  PRINT : PRINT : HTAB 16: PRINT D$;"I/O ERRO
     R": GOSUB 9090
9035  IF  PEEK (222) = 10 THEN  PRINT : PRINT : HTAB 14: PRINT D$;"FILE LO
     CKED": GOSUB 9090
9040  IF  PEEK (222) = 16 THEN  PRINT : PRINT : HTAB 14: PRINT D$;"SYNTAX
     ERROR": GOSUB 9090
9050  IF  PEEK (222) = 13 THEN  PRINT : PRINT : HTAB 12: PRINT D$;"FILE TY
     PE MISMATCH": GOSUB 9090
9080  NORMAL
9085  GOTO 9100
9090  FOR E = 1 TO 1000: NEXT : RETURN
9100  GOTO 610
10000  REM ***   SAVE   ***
10010  POKE 34,0: POKE 33,40: HOME
10020  HTAB 18: INVERSE : PRINT "SAVE"
10030  VTAB 5: HTAB 14: INVERSE : PRINT "RETURN": NORMAL : VTAB 5: HTAB 21
     : PRINT "= MENU"
10040  PRINT : PRINT : PRINT "IF YOU'RE CONTINUIN' THE ";: INVERSE : PRINT
     "SAVE";: NORMAL : PRINT "-PART OF": PRINT : PRINT "THIS PROGRAM, IT W
     ILL DESTROY THE FIRST": PRINT : PRINT "FILE ON YOUR DISK CALLED 'HELL
     O', IN-"
10050  PRINT : PRINT "DEPENDENT OF IT'S LOCKED OR NOT. INSTEAD": PRINT "OF
      THAT, THIS PROGRAM WILL SELF-SAVE AND": PRINT "-LOCK. IF YOU'RE READ
     Y - HIT 'Y'...": PRINT
10060  GET E$: IF E$ = "Y" THEN 10070
10062  IF E$ =  CHR$ (13) THEN 610
10065  GOTO 10060
10070 D$ =  CHR$ (13) +  CHR$ (4): PRINT D$;"UNLOCK HELLO": PRINT D$;"DELE
     TE HELLO": PRINT D$;"SAVE HELLO": PRINT D$;"LOCK HELLO": GOTO 610
```

# Auflösung des CPU-Rätsels aus Heft Nr. 6

Wenn Sie die Fragen dieses Rätsels richtig beantwortet haben, ergibt sich folgende Zahl:

- Welche Aufgabe hat ein Videocontroller? 3 = Bildaufbau in einem Computer
- Bedeutung von CP/M? 5 = Diskettenbetriebssystem
- Was ist ein Sprite? 7 = Durch Hardware belegte Figur
- Welche Funktion erfüllt ein Paddle? 10 = Drehregler für analoge Eingaben
- Welche Computersprache verwendet zum Zeichnen eine Schildkröte? 14 = Logo
- Wozu dient ein RESET? 17 = Zum Rücksetzen des Computers
- Welcher Teil eines Computers wird mit Akkumulator bezeichnet? 20 = Hauptregister des Prozessors
- Was ist BIOS? 23 = Teil eines bekannten Betriebssystems
- Ein Netzwerk ist? 25 = Die Zusammenschaltung von mehreren Computern

3 + 5 + 7 + 10 + 14 + 17 + 20 + 23 + 25 = 124

# 2. Elektronik- & Computertage Saar

## 2. Verkaufs- und Informationsmesse für Elektronik- und Microcomputer vom 6.–8. September 1985 in der Kongreßhalle Saarbrücken

Mit über 60 Ausstellern und fast 7000 Besuchern waren die 1. Elektronik- & Computertage Saar für alle Beteiligten ein großer Erfolg.
Nach der Eröffnung durch den Schirmherrn, dem saarländischen Landtagspräsidenten Albrecht Herold, Vertretern der Ministerien sowie Presse, Rundfunk und Fernsehen, bot sich den Besuchern auf einer Ausstellungsfläche von fast 1100 qm ein repräsentativer Querschnitt durch die gesamte Elektronik- und Microcomputerbranche.

Einen sehr hohen Besucheranteil stellten Vertreter von Handwerk- und Industriebetrieben aus dem Saarland und den benachbarten Regionen dar. Besonders erfreulich war die Tatsache, daß viele Besucher aus Frankreich, Luxemburg und sogar aus Belgien die Elektronik- & Computertage Saar besucht haben.
An diesem positiven Ergebnis hatte sicherlich die Präsens von namhaften Firmen wie APPLE, EPSON, FUJITSU, HP, IBM, KONTRON, OLIVETTI, TOSHIBA, TRIUMPH ADLER, COMMODORE und WANG, um nur einige wenige zu nennen, einen großen Anteil.
Daß diese Firmen auch dieses Jahr ihr Kommen, teilweise mit erheblicher Standvergrößerung, wieder zugesagt haben, ist ein Grundstein für ein noch besseres Resultat.

Neben den kommerziellen Rechnern findet der Interessent eine große Auswahl an Zubehör und Fachliteratur. Aber auch die Elektronik mit modernen Kommunikationstechniken und der Video- und Audiobereich gehören zum Ausstellungsschwerpunkt. Im Foyer West der Kongreßhalle finden unsere jüngeren Besucher ihr „Heimcomputerparadies", mit allem Zubehör, was das Herz begehrt.
Die Elektronik- & Computertage Saar werden auch 1985 nach dem bereits bewährten Konzept mit folgenden Schwerpunkten ausgerichtet:
– Kommerzielle Hard- und Software aller marktführenden Hersteller für Handel, Handwerk, Industrie und Dienstleistung!
– Ständige Vorführungen von Standard-Software, wie Finanzbuchhaltung, Lohn- und Gehaltsabrechnung, Lager- und Auftragsverwaltung sowie der modernen Textverarbeitung für den Klein- und Mittelbetrieb!

– Besondere Programmprodukte für freiberuflich Tätige wie Rechtsanwälte, Ärzte, Steuerberater, Architekten usw.!
– Fachvorträge durch Referenten namhafter Firmen und dem Institut für Informatik in Saarbrücken!
– Große Auswahl an Fachliteratur!
– Audio- und Video sowie moderne Kommunikationstechniken!
– Umfangreicher Zubehörhandel vom Laserdrucker bis zur einfachen Diskette!
– Heimcomputer und Zubehör in allen Preisklassen für unsere jüngeren Messebesucher!
– Meßgeräte, Elektronikteile und Zubehör für Profis und Bastler!
– Besondere Vorführungen im Bereich Amateurfunk, wie Meteosatempfang, Funkfernschreiben per Computer usw., durch den Deutschen Amateur Radio Club, Ortsverband Völklingen.

# Ein Super-Disassembler für den TI-99

**Zum Programmieren in Maschinensprache benötigt man neben einem Assembler und einem Monitorprogramm auch einen Disassembler. Einen solchen stellt das folgende Programm dar, das um viele Besonderheiten erweitert wurde. Es läuft auf den TI-99 mit Extended Basic, Diskette und 32K-Speichererweiterung.**
Disassembler werden verwendet, um z. B. das Betriebssystem eines Computers zu analysieren oder Fehler in eigenen Maschinenprogrammen aufzuspüren. Der abgedruckte Disassembler hat aber einige Zusatzfunktionen, die meist fehlen, obwohl diese sehr wichtig sind:
– Relative Sprünge und CRU-Bit-Befehle werden auch für negative Werte korrekt disassembliert, z. B. JOC $-8 oder TB -20. Dabei wird bei Relativsprüngen die absolute Adresse des Sprungziels berechnet.
– Der Pseudo-Befehl NOP (No Operation) wird erkannt.
– Die Befehle IDLE, RSET, CKON, CKOF und LREX, die zwar der TMS9900 versteht, die aber auf dem TI-99 nicht verwendet werden, werden ebenfalls disassembliert und mit einem Fragezeichen angezeigt, z. B. "IDLE?".
– Die wichtigste Besonderheit: Adressen im CPU-RAM von ›8300 bis ›83FF sowie der Sprung in den GPL-Interpreter NEXT (›0070) werden statt eines Zahlenwertes als Labels ausgegeben. Z. B. ist der Befehl SB §STATUS-,§JOYSTX erheblich leichter zu verstehen als SB §›837C,§›8377.
Zur Programmbedienung: Sobald es mit RUN gestartet wird, erscheint eine übersichtliche Eingabemaske. Hier geben Sie die Start- und die Endadresse des zu disassemblierenden Bereichs ein. Diese Eingaben können Sie dezimal oder hexadezimal machen. Im letzteren Fall müssen Sie das Hex-Symbol "›" voranstellen. Außerdem können Sie wählen, ob das Listing zusätzlich auf dem Drucker oder auf Diskette ausgegeben werden soll.

Kurz darauf beginnt das Programm mit dem Disassemblieren. Die Ausgaben haben auf dem Drucker und dem Bildschirm das gleiche Format, nur daß auf dem Drucker mehr Zeichen in jeder Zeile darstellbar sind.

```
0AA8  02E0  LWPI  GPLWS
      83E0
0AAC  B80E  AB    R14,§TIMER
      8379
0AB0  C320  .MOV  §UDFINT,R12
      83C4
0AB4  1301  JEQ   ›0AB8
0AB6  069C  BL    *R12
0AB8  04C8  CLR   R8
0ABA  02E0  LWPI  INTWS
      83C0
0ABE  0380  RTWP
```

1. Spalte: Speicheradresse
2. Spalte: Inhalt der Adresse
3. Spalte: Assembler-Befehl
4. Spalte: Zugehörige Operanden

Folgende Tabellen werden statt der entsprechenden Adressen angezeigt:

›0070 NEXT = Sprung in den GPL-Interpreter
›2008 NUMASG = Numerischen Parameter zuweisen
›200C NUMREF = Numerischen Parameter holen
›2010 STRASG = Stringparameter zuweisen
›2014 STRREF = Stringparameter holen
›2018 XMLLNK = ROM-Routinen aufrufen
›201C KSCAN = Tastaturabfrage
›2020 VSBW = Ein Byte in den Video-RAM schreiben
›2024 VMBW = Mehrere Bytes in den Video-RAM schreiben
›2028 VSBR = Ein Byte aus dem Video-RAM lesen
›202C VMBR = Mehrere Bytes aus dem Video-RAM lesen
›2030 VMTR = VDP-Register schreiben
›2034 ERR = Fehlermeldung ausgeben
›8300 PAD = CPU-RAM-Pad
›831E BASCMD = Zeiger auf gerade bearbeiteten Basic-Befehl
›8322 ERRCOD = Fehlercode bei Rückkehr aus Maschinenprogramm
›832E ACTLIN = Zeiger auf gerade bearbeitete Basic-Zeile
›8330 LSTLIN = Zeiger auf Ende der Zeilenliste
›8332 FSTLIN = Zeiger auf Beginn der Zeilenliste
›833A SUBLST = Zeiger auf Liste der CALL-Unterprogramme
›833C PABLST = Zeiger auf Liste der PAB-Blöcke
›833E VARLST = Zeiger auf Liste der Variablen
›8340 VDPSTR = Zeiger auf freien Video-RAM für Strings
›8344 RUNFLG = Basic-Run-Flag (0 oder 255)
›8345 BASFLG = Basic-Flag-Byte (TRACE, PROTECTION, NUM etc.)
›834A FAC = Fließkomma-Akku 1
›8354 OVLERR = Fehlerflag bei Fließkomma-Operationen
›835C ARG = Fließkomma-Akku 2
›836E VSPTR = Value Stack Pointer
›8370 VHIMEM = Höchste nutzbare Adresse im Video-RAM
›8372 DATSTK = Zeiger auf GPL-Daten-Stack
›8373 SUBSTK = Zeiger auf GPL-Unterprogramm-Stack
›8374 KBOARD = In CALL KEY gewählter Tastaturmodus
›8375 ASCII = Wert der gedrückten Taste bei SCAN/KSCAN
›8376 JOYSTY = Y-Koordinate des Joysticks
›8377 JOYSTX = X-Koordinate des Joysticks
›8378 RANDOM = Zufallszahlen
›8379 INTCNT = Interruptgesteuerter Counter
›837A SPRMOT = Höchste Nummer der Sprites, die bewegt werden können
›837B VSTAT = VDP-Status-Register
›837C STATUS = GPL-Statusbyte
›8384 XHIMEM = Höchste nutzbare Adresse der Memory Expansion
›8386 XHFREE = Zeiger auf höchste freie Adresse der Memory Expansion
›83C0 INTWS = Interrupt-Workspace
›83C2 IFLAG = Interrupt-Flag
›83C4 UDFINT = User-defined Interrupt
›83CC SNDLST = Zeiger auf Soundliste für automatische Tonerzeugung
›83CE SNDBYT = Gerade gespieltes Soundbyte
›83D0 GRMSRC = GROM-Suchadresse
›83D2 ROMSRC = ROM-Suchadresse
›83D4 V1COPY = Kopie des VDP-Registers 1
›83D6 CLSCNT = Counter für Clearscreen
›83D8 SCANRT = Rücksprungadresse von SCAN/KSCAN
›83E0 GPLWS = GPL-Workspace

```
100 !  ******************
110 ! ********************
120 ! *                  *
130 ! *   DISASSEMBLER   *
140 ! *                  *
150 ! ********************
160 !  ******************
170 !
180 ! Für den TI-99 mit Extended Basic
190 !
200 ! (c) Martin Kotulla 1985
210 !
220 ! ----------------------------------------------
230 !
240 DATA "SZC ","SZCB","S   ","SB  ","C   ","CB  "
250 DATA "A   ","AB  ","MOV ","MOVB","SOC ","SOCB"
260 DATA "JMP ","JLT ","JLE ","JEQ ","JHE ","JGT "
270 DATA "JNE ","JNC ","JOC ","JNO ","JL  ","JH  "
280 DATA "JOP ","SBO ","SBZ ","TB  "
290 DATA "COC ","CZC ","XOR "
300 DATA "LDCR","STCR"
```

```
310 DATA "SRA ","SRL ","SLA ","SRC "
320 DATA "BLWP","B   ","X   ","CLR ","NEG ","INV "
330 DATA "INC ","INCT","DEC ","DECT","BL  ","SWPB"
340 DATA "SETO","ABS "
350 DATA "LI  ","AI  ","ANDI","ORI ","CI  "
360 ! Labels für CPU-RAM im Bereich >8300 bis >83FF ****************************
370 DATA 0,PAD,30,BASCMD,34,ERRCOD,46,ACTLIN,48,LSTLIN,50,FSTLIN,58,SUBLST
380 DATA 60,PABLST,62,VARLST,64,VDPSTR,68,RUNFLG,69,BASFLG,74,FAC
390 DATA 84,OVLERR,92,ARG,110,VSPTR,112,VHIMEM,114,DATSTK,115,SUBSTK
400 DATA 116,KBOARD,117,ASCII,118,JOYSTY,119,JOYSTX,120,RANDOM,121,INTCNT
410 DATA 122,SPRMOT,123,VSTAT,124,STATUS,132,XHIMEM,134,XHFREE,192,INTWS
420 DATA 194,IFLAG,196,UDFINT,204,SNDLST,206,SNDBYT,208,GRMSRC,210,ROMSRC
430 DATA 212,V1COPY,214,CLSCNT,216,SCANRT,224,GPLWS
440 ! Labels für Assembler-Hilfsprogramme im Low Memory von >2000 - >24F4 ******
450 DATA NUMASG,NUMREF,STRASG,STRREF,XMLLNK,KSCAN,VSBW,VMBW,VSBR,VMBR,VWTR,ERR
460 ! Definitionen und Daten einlesen ****************************************
470 DIM FM1$(11),FM2$(15),FM3$(2),FM4$(1),FM5$(3)
480 DIM FM6$(13),FM8$(4),LABEL$(1,255)
490 FOR I=0 TO 11 :: READ FM1$(I):: NEXT I
500 FOR I=0 TO 15 :: READ FM2$(I):: NEXT I
510 READ FM3$(0),FM3$(1),FM3$(2)
520 READ FM4$(0),FM4$(1)
530 READ FM5$(0),FM5$(1),FM5$(2),FM5$(3)
540 FOR I=0 TO 13 :: READ FM6$(I):: NEXT I
550 FOR I=0 TO 4 :: READ FM8$(I):: NEXT I
560 HX$="0123456789ABCDEF"
570 DEF HEX$(X)=SEG$(HX$,INT(X/16)+1,1)&SEG$(HX$,X-INT(X/16)*16+1,1)
580 DEF HEXW$(X)=HEX$(INT(X/256))&HEX$(X-INT(X/256)*256)
590 DEF DEZ(X$)=(POS(HX$,SEG$(X$,1,1),1)-1)*16+POS(HX$,SEG$(X$,2,1),1)-1
600 DEF DEZW(X$)=DEZ(SEG$(X$,1,2))*256+DEZ(SEG$(X$,3,2))
610 DEF COMPL2=I+65536*(I>32767)! Zweierkomplement einer Integerzahl
620 ! Titelbild ***************************************************************
630 CALL CHAR(42,"AA55AA55AA55AA55",91,"7E8199A1A199817E")
640 CALL CHAR(136,RPT$("0",8)&"FF")
650 CALL CHAR(128,"DFBF60C0919B9591918080C060BFDF")
660 CALL CHAR(130,"FBFD0603010101495161514903O6FDFB")
670 CALL CLEAR :: CALL HCHAR(2,4,42,26)
680 CALL VCHAR(3,4,42,5):: CALL VCHAR(3,29,42,5)
690 CALL MAGNIFY(3):: CALL SPRITE(#1,128,2,28,60)
700 DISPLAY AT(5,11)SIZE(12):"DISASSEMBLER"
710 CALL HCHAR(8,4,42,26)
720 DISPLAY AT(12,4):CHR$(91)&"  MARTIN KOTULLA 1985"
730 CALL HCHAR(13,6,136,22)
740 DISPLAY AT(15,1):"- STARTADRESSE?": :"- ENDADRESSE?"
750 DISPLAY AT(19,1):"- DRUCKERAUSGABE?  N"
760 ACCEPT AT(15,20)SIZE(5)VALIDATE(HX$,">"):ST$
770 IF ASC(ST$)=62 THEN START=DEZW(SEG$(ST$,2,4))ELSE START=VAL(ST$)
780 START=INT(START/2)*2
790 ACCEPT AT(17,20)SIZE(5)VALIDATE(HX$,">"):EN$
800 IF ASC(EN$)=62 THEN ENDE=DEZW(SEG$(EN$,2,4))ELSE ENDE=VAL(EN$)
810 ENDE=INT(ENDE/2)*2
820 ACCEPT AT(19,20)SIZE(-1)VALIDATE("JN"):FILE$
830 IF FILE=1 THEN CLOSE #1 :: FILE=0
840 IF FILE$="N" THEN FILE=0 :: GOTO 890
850 FILE=1
860 DISPLAY AT(21,1):"- DRUCKERNAME? RS232"
870 ACCEPT AT(21,16)SIZE(-12)VALIDATE(UALPHA,DIGIT,".="):DRUCK$
880 OPEN #1:DRUCK$
890 CALL CLEAR :: CALL CHARSET :: CALL DELSPRITE(#1)
900 FOR I=START TO ENDE STEP 2
910 CALL PEEK(COMPL2,HIBYTE,LOBYTE)
920 WORD=HIBYTE*256+LOBYTE
930 IF WORD<32768 THEN NEGWORD=WORD ELSE NEGWORD=WORD-65536
940 OUT$=HEXW$(I)&" "&HEXW$(WORD)&" "
950 STARTCMD=I
```

```
960 ! Pseudo-Befehl NOP ***
970 IF WORD=4096 THEN OUT$=OUT$&"NOP" :: GOTO 1270
980 ! Format 1 ************
990 IF WORD>16383 THEN 1480
1000 ! Format 2a ***********
1010 IF WORD>4095 AND WORD<7424 THEN 1570
1020 ! Format 2b ***********
1030 IF WORD>7423 AND WORD<8192 THEN 1640
1040 ! Format 3 ************
1050 IF WORD>8191 AND WORD<11264 THEN 1690
1060 ! Format 4 ************
1070 IF WORD>12287 AND WORD<14336 THEN 1770
1080 ! Format 5 ************
1090 IF WORD>2047 AND WORD<3072 THEN 1850
1100 ! Format 6 ************
1110 IF WORD>1023 AND WORD<2048 THEN 1910
1120 ! Format 7 ************
1130 IF WORD=832 OR WORD=864 OR WORD=896 THEN 1970
1140 IF WORD=928 OR WORD=960 OR WORD=992 THEN 1970
1150 ! Format 8a ***********
1160 IF WORD>511 AND WORD<672 THEN 2050
1170 ! Format 8b ***********
1180 IF WORD=736 OR WORD=768 THEN 2120
1190 ! Format 8c ***********
1200 IF WORD>671 AND WORD<720 THEN 2160
1210 ! Format 9a ***********
1220 IF WORD>11263 AND WORD<12288 THEN 2210
1230 ! Format 9b ***********
1240 IF WORD>14335 AND WORD<16384 THEN 2280
1250 ! DATA-Befehle ********
1260 OUT$=OUT$&"DATA >"&HEXW$(WORD)
1270 CALL HCHAR(24,1,ASC(OUT$))
1280 CALL HCHAR(24,2,ASC(SEG$(OUT$,2,1)))
1290 IF LEN(OUT$)<31 THEN PRINT SEG$(OUT$,3,255):: GOTO 1360
1300 PRINT SEG$(OUT$,3,28)
1310 CALL HCHAR(23,31,ASC(SEG$(OUT$,31,1)&"    "))
1320 CALL HCHAR(23,32,ASC(SEG$(OUT$,32,1)&"    "))
1330 IF LEN(OUT$)<33 THEN 1360
1340 PRINT :: FOR K=33 TO LEN(OUT$)
1350 CALL HCHAR(23,K-17,ASC(SEG$(OUT$,K,1))):: NEXT K
1360 IF FILE=1 THEN PRINT #1:OUT$
1370 X1=STARTCMD+2 :: IF X1>32767 THEN X1=X1-65536
1380 X2=COMPL2
1390 FOR J=X1 TO X2 STEP 2
1400 CALL PEEK(J,A,B)
1410 PRINT TAB(4);HEXW$(A*256+B)
1420 IF FILE=1 THEN PRINT #1:TAB(6);HEXW$(A*256+B)
1430 NEXT J
1440 NEXT I
1450 PRINT :: DISPLAY BEEP:"FERTIG-BITTE SPACE DRUECKEN!"
1460 CALL KEY(0,K,S):: IF K<>32 THEN 1460 ELSE 630
1470 ! FORMAT 1 ***********************************************************
1480 OUT$=OUT$&FM1$(INT(WORD/4096)-4)&" "
1490 DESTINATION=15 AND(INT(WORD/64))
1500 SOURCE=15 AND NEGWORD
1510 TYPEDEST=3 AND(INT(WORD/1024))
1520 TYPESOURCE=3 AND(INT(WORD/16))
1530 CALL DECODE(SOURCE,TYPESOURCE,I,"S",OUT$)! SUB "OPERANDEN DEKODIEREN"
1540 CALL DECODE(DESTINATION,TYPEDEST,I,"D",OUT$)! SUB "OPERANDEN DEKODIEREN"
1550 GOTO 1270
1560 ! FORMAT 2a **********************************************************
1570 OUT$=OUT$&FM2$(INT(WORD/256)-16)&" "
1580 DISPLACE=LOBYTE*2
1590 IF DISPLACE>255 THEN DISPLACE=DISPLACE-512
```

```
1600 JUMPADDR=I+DISPLACE+2
1610 IF JUMPADDR=112 THEN OUT$=OUT$&"NEXT" :: GOTO 1270
1620 OUT$=OUT$&">"&HEXW$(JUMPADDR):: GOTO 1270
1630 ! FORMAT 2b ************************************************************
1640 OUT$=OUT$&FM2$(INT(WORD/256)-16)&" "
1650 WORD=WORD AND 255
1660 IF WORD<128 THEN OUT$=OUT$&STR$(WORD)ELSE OUT$=OUT$&STR$(WORD-256)
1670 GOTO 1270
1680 ! FORMAT 3 *************************************************************
1690 OUT$=OUT$&FM3$(INT(WORD/1024)-8)&" "
1700 TYPESOURCE=INT(WORD/16)AND 3
1710 SOURCE=NEGWORD AND 15
1720 CALL DECODE(SOURCE,TYPESOURCE,I,"S",OUT$)! SUB "OPERANDEN DEKODIEREN"
1730 DESTINATION=INT(WORD/64)AND 15
1740 OUT$=OUT$&"R"&STR$(DESTINATION)
1750 GOTO 1270
1760 ! FORMAT 4 *************************************************************
1770 OUT$=OUT$&FM4$(INT(WORD/1024)-12)&" "
1780 TYPESOURCE=INT(WORD/16)AND 3
1790 SOURCE=NEGWORD AND 15
1800 CALL DECODE(SOURCE,TYPESOURCE,I,"S",OUT$)! SUB "OPERANDEN DEKODIEREN"
1810 CRULINE=INT(WORD/64)AND 15
1820 OUT$=OUT$&STR$(CRULINE)
1830 GOTO 1270
1840 ! FORMAT 5 *************************************************************
1850 OUT$=OUT$&FM5$(HIBYTE-8)&" "
1860 REGISTER=LOBYTE AND 15
1870 SHIFTCOUNT=INT(LOBYTE/16)AND 15
1880 OUT$=OUT$&"R"&STR$(REGISTER)&","&STR$(SHIFTCOUNT)
1890 GOTO 1270
1900 ! FORMAT 6 *************************************************************
1910 OUT$=OUT$&FM6$(INT(WORD/64)-16)&" "
1920 TYPESOURCE=INT(WORD/16)AND 3
1930 SOURCE=WORD AND 15
1940 CALL DECODE(SOURCE,TYPESOURCE,I,"D",OUT$)! SUB "OPERANDEN DEKODIEREN"
1950 GOTO 1270
1960 ! FORMAT 7 *************************************************************
1970 IF WORD=832 THEN OUT$=OUT$&"IDLE?"
1980 IF WORD=864 THEN OUT$=OUT$&"RSET?"
1990 IF WORD=896 THEN OUT$=OUT$&"RTWP"
2000 IF WORD=928 THEN OUT$=OUT$&"CKON?"
2010 IF WORD=960 THEN OUT$=OUT$&"CKOF?"
2020 IF WORD=992 THEN OUT$=OUT$&"LREX?"
2030 GOTO 1270
2040 ! FORMAT 8a ************************************************************
2050 OUT$=OUT$&FM8$(INT(WORD/32)-16)&" "
2060 REGISTER=NEGWORD AND 15
2070 OUT$=OUT$&"R"&STR$(REGISTER)
2080 CALL DEEK(I,IMMEDIATE)
2090 OUT$=OUT$&",>"&HEXW$(IMMEDIATE)
2100 GOTO 1270
2110 ! FORMAT 8b ************************************************************
2120 CALL DEEK(I,IMMEDIATE)
2130 IF WORD=736 THEN OUT$=OUT$&"LWPI >"&HEXW$(IMMEDIATE):: CALL LABEL(I,OUT$)::
 GOTO 1270
2140 OUT$=OUT$&"LIMI "&STR$(IMMEDIATE):: GOTO 1270
2150 ! FORMAT 8c ************************************************************
2160 IF WORD<704 THEN OUT$=OUT$&"STWP" ELSE OUT$=OUT$&"STST"
2170 REGISTER=NEGWORD AND 15
2180 OUT$=OUT$&" R"&STR$(REGISTER)
2190 GOTO 1270
2200 ! FORMAT 9a ************************************************************
2210 OUT$=OUT$&"XOP "
2220 SOURCE=NEGWORD AND 15
2230 TYPESOURCE=INT(WORD/16)AND 3
```

```
2240 CALL DECODE(SOURCE,TYPESOURCE,I,"S",OUT$)! SUB "OPERANDEN DEKODIEREN"
2250 DESTINATION=INT(WORD/64)AND 15
2260 OUT$=OUT$&STR$(DESTINATION):: GOTO 1270
2270 ! FORMAT 9b ************************************************************
2280 IF WORD<15360 THEN OUT$=OUT$&"MPY   " ELSE OUT$=OUT$&"DIV   "
2290 SOURCE=NEGWORD AND 15
2300 TYPESOURCE=INT(WORD/16)AND 3
2310 CALL DECODE(SOURCE,TYPESOURCE,I,"S",OUT$)! SUB "OPERANDEN DEKODIEREN"
2320 DESTINATION=INT(WORD/64)AND 15
2330 OUT$=OUT$&"R"&STR$(DESTINATION):: GOTO 1270
2340 ! *********************************************************************
2350 ! UNTERPROGRAMME *******************************************************
2360 ! *********************************************************************
2370 ! Adressierungsart dekodieren ------------------------------------------
2380 SUB DECODE(OPERAND,TYPE,I,SD$,OUT$)
2390 DEF COMPL2=I+65536*(I>32767)
2400 ON TYPE+1 GOTO 2420,2440,2460,2550
2410 ! *-* WORKSPACE REGISTER *-*
2420 OUT$=OUT$&"R"&STR$(OPERAND):: GOTO 2560
2430 ! *-* WORKSPACE REGISTER INDIRECT *-*
2440 OUT$=OUT$&"*R"&STR$(OPERAND):: GOTO 2560
2450 ! *-* SYMBOLIC MEMORY/INDEXED MEMORY *-*
2460 I=I+2 :: CALL PEEK(COMPL2,A,B)
2470 HX$="0123456789ABCDEF"
2480 HX1$=SEG$(HX$,INT(A/16)+1,1)&SEG$(HX$,A-INT(A/16)*16+1,1)
2490 HX2$=SEG$(HX$,INT(B/16)+1,1)&SEG$(HX$,B-INT(B/16)*16+1,1)
2500 IF OPERAND<>0 THEN 2520
2510 OUT$=OUT$&CHR$(64)&">"&HX1$&HX2$ :: CALL LABEL(I,OUT$):: GOTO 2560
2520 OUT$=OUT$&CHR$(64)&">"&HX1$&HX2$ :: CALL LABEL(I,OUT$)
2530 OUT$=OUT$&"(R"&STR$(OPERAND)&")" :: GOTO 2560
2540 ! *-* WORKSPACE REGISTER INDIRECT AUTO-INCREMENT *-*
2550 OUT$=OUT$&"*R"&STR$(OPERAND)&"+"
2560 IF SD$="S" THEN OUT$=OUT$&","
2570 SUBEND
2580 ! Double Peek (16-Bit-Wort lesen) ---------------------------------------
2590 SUB DEEK(I,C)
2600 I=I+2 :: COMPL2=I+65536*(I>32767)
2610 CALL PEEK(COMPL2,A,B)
2620 C=A*256+B :: SUBEND
2630 ! Adresse durch Label ersetzen -----------------------------------------
2640 SUB LABEL(I,OUT$)
2650 IF FLG THEN 2690 ELSE FLG=-1
2660 RESTORE 370 :: DIM LABEL$(1,255)
2670 FOR J=0 TO 40 :: READ ADR,LABEL$(0,ADR):: NEXT J
2680 FOR J=0 TO 11 :: ADR=J*4+8 :: READ LABEL$(1,ADR):: NEXT J
2690 CALL PEEK(I,A,B)
2700 IF A*256+B=112 THEN OUT$=SEG$(OUT$,1,LEN(OUT$)-5)&"NEXT" :: SUBEXIT
2710 IF A<>131 THEN 2740
2720 IF LABEL$(0,B)<>"" THEN OUT$=SEG$(OUT$,1,LEN(OUT$)-5)&LABEL$(0,B)
2730 SUBEXIT
2740 IF A<>32 THEN 2760
2750 IF LABEL$(1,B)<>"" THEN OUT$=SEG$(OUT$,1,LEN(OUT$)-5)&LABEL$(1,B)
2760 SUBEND
2770 END ! *************************************************************
```

* * * * * * * * * * * * * * *

## Computer-Messen '85

| | |
|---|---|
| 30. 8.– 8. 9. 85 | Internationale Funkausstellung Berlin |
| 24. 9.–27. 9. 85 | CAMP/Computer Graphics/Berlin |
| Oktober | EL-FA Elektrotechnische Fachausstellung/Berlin |
| 9. 10.–13. 10. 85 | Hobby-Elektronik 85/ Stuttgart |
| 9. 10.–14. 10. 85 | Frankfurter Buchmesse |
| 17. 10.–20. 10. 85 | CFA – Commodore Fachausstellung/Frankfurt |
| 28. 10.– 1. 11. 85 | SYSTEMS 85/München |
| 12. 11.–16. 11. 85 | PRODUCTRONICA Intern. Fachmesse für die Fertigung in der Elektronik/München |

## Computermessen im Ausland

| | |
|---|---|
| September 85 | SICOB/Paris |
| 18. 9.–22. 9. 85 | PCW/London |
| 26. 9.– 2. 10. 85 | Data – Internationale Messe für Computer und Büro-Rationalisierung |
| Oktober 85 | COMDEX/Las Vegas, USA |
| November 85 | ELKOM – Internationale Elektronik-Ausstellung/ Helsinki |
| 5. 11.–8. 11. 85 | ELFACK – Internationale Elektronische Messe/ Göteborg |

**Herausgeber des auf Seite 17 vorgestellten Mailbox-Buches ist Claus Warneke, Kirchhuchtinger Landstr. 243, 2800 Bremen 66**

# Tips und Tricks für den CPC-464

```
100 n=12345
110 a$=DEC$((n,"######.##")
120 PRINT a$
130 REM  FORMATIEREN VON ZAHLEN FUER STRINGS MIT DEM CPC464
140 REM  ==================================================
150 REM
160 REM  Das Formatieren von Zahlen bedeutete immer dann
170 REM  einen Programmieraufwand, wenn die Zahlen in
180 REM  Strings umzuwandeln waren, im Gegensatz zur
190 REM  formatierten Ausgabe in der PRINT-Anweisung
200 REM  mittels PRINT USING. Der CPC464 besitzt einen
210 REM  entsprechenden Formatierbefehl fuer Zahlen.
220 REM  Allerdings enthaelt der Aufruf einen Schoenheits-
230 REM  fehler, vermutlich der Grund, warum diese Funktion
240 REM  nicht im Benutzer-Handbuch beschrieben wurde.
250 REM
260 REM  Der Aufruf geht wie folgt:
270 REM  DEC$((<Ausdruck>,"Formatierungszeichen")
280 REM  wobei zu beachten ist, dasz
290 REM  1. hinter DEC$ eine zusaetzliche Klammer (der
300 REM     "Schoenheitsfehler") stehen musz, die auch
310 REM     nicht mehr zugemacht werden darf,
320 REM  2. im Formatierungsstring keine BLANKS, d.h. also
330 REM     nur die Zeichen  "#","+","-","." und "^" ent-
340 REM     halten sein duerfen.
```

# TRONIC-SOFTWARE-SERVICE

**Atari,** Mastermind/Schlangenkrieg, Kassette 10,50 DM, Diskette 19,50 DM, Bestell-Nr. I 41

Tank-Battle/Oil Panic, Kassette 12,50 DM, Bestell-Nr. I 51

Startup/Zeilen-Split/Tomstone-City, Diskette 19,50 DM, Bestell-Nr. I 71

Painter/Hardcopy, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. I 81

The Big Quest/Fünf gewinnt, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. I 91

Splitt, Kassette 11,– DM, Diskette 17,50 DM, Bestell-Nr. I 101

Ski/Mutation, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. I 121

Super Miner/Diamonds, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. I 22

Donkey Kong/Kerzenheinz, Kassette 16,50 DM, Diskette 19,50 DM, Bestell-Nr. I 32.

**TI-99,** Drei-Kronen-Spiel/Zahlenputzen, Kassette 8,50 DM, Bestell-Nr. T 41

Karl der Käfer/Alien-Landing, Kassette 14,50 DM, Bestell-Nr. T 51

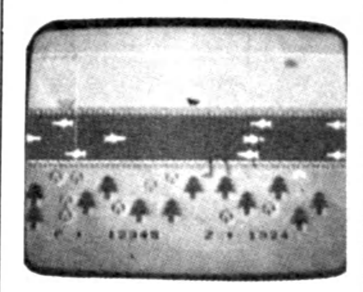
Jack the Digger I/Noah 2099, Kassette 14,50 DM, Bestell-Nr. T 61

Lift Bär/ASC II DEF Teil 1, Kassette 14,– DM, Bestell-Nr. T 71

Maya/ASC II DEF Teil 2, Kassette 14,50 DM, Bestell-Nr. T 81

ASC II DEF Teil 1 + 2, Diskette 19,50 DM, Bestell-Nr. T 881

Transfer/Silverspar, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. T 91



Mother Duck/Screen Designer, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. T 101

Cave Man, Kassette 8,– DM, Diskette 11,50 DM, Bestell-Nr. T 121

Moon Race/Frogger/Slicks, Kassette 19,50 DM, Diskette 25,50 DM, Bestell-Nr. T 22

Panzerschlacht, Kassette 8,– DM, Diskette 11,50 DM, Bestell-Nr. T 32.

**Commodore 64,** Mauern/Widerstand, Kassette 8,– DM, Diskette 15,– DM, Bestell-Nr. C 41

Space-Comets/Erdspalte/Sprite-Data, Kassette 15,– DM, Diskette 23,50 DM, Bestell-Nr. C 51

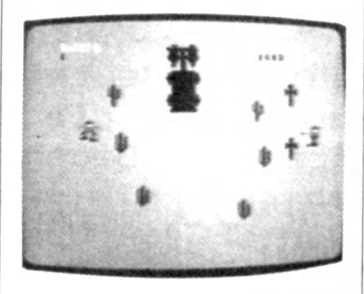
Autostart/Bestellschein/Roadpainter, Kassette 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 61

Hardcopy/Space-Fighter/Data-Generator, Kassette 15,50 DM, Diskette 19,50 DM, Bestell-Nr. C 71

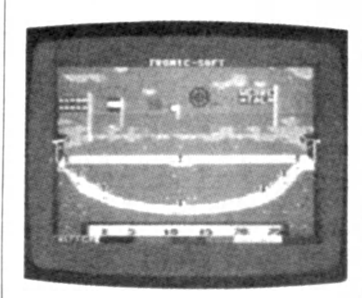
Monster-Attack/Block-Painter/Epson-Drucker, Kassette 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 81

Projekt/Datenbank, Kassette 16,– DM, Diskette 23,50 DM, Bestell-Nr. C 91

Spiders/The Basic, Kassette 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 101



High Noon/Skeet/Grafik-Designer, Kassette 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 121




Painter/Star-Baddle/Editor, Kassette 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 22

Wüstenrallye/Jet-Pac/Black Moore Castle Kassette 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 32.

**VC-20,** Bestellschein/Glükky, Kassette 8,50 DM, Diskette 15,– DM, Bestell-Nr. V 61

Multigraph/All Rammer, Kassette 11,– DM, Diskette 15,50 DM, Bestell-Nr. V 71

Zyklo/Meteorit, Kassette 11,– DM, Diskette 19,50 DM, Bestell-Nr. V 81

Garten/Schloß Gruselstein, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 91

Fressman/Outlaw, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 101

Prost/Buffalo Bill, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 121

Joy Man/Powerpack, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 22

Der rasende Malocher/Frankie goes to Pharao, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 32.

**Apple II,** Wilder Westen/Karambolage/Maskengenerator, Diskette 19,50 DM, Bestell-Nr. A 41

Music-Maker/Mission: Adler/Disk-Kata-log, Diskette 19,50 DM, Bestell-Nr. A 51

Snake/Super Datei/Shapetables, Diskette 19,50 DM, Bestell-Nr. A 61

Library/Fight, Diskette 19,50 DM, Bestell-Nr. A 71

Reversal/Disk-Menue-Generator, Diskette 19,50 DM, Bestell-Nr. A 81

Diamonds/Hilfsprogramm, Diskette 19,50 DM, Bestell-Nr. A 91

Tic-Tac-Toe/Jumper, Diskette 19,50 DM, Bestell-Nr. A 101

Donovan/Basic-Konverter, Diskette 19,50 DM, Bestell-Nr. A 121

Funktionstasten/Painter/Bowling, Diskette 19,50 DM, Bestell-Nr. A 22

Thunder/Castle of Doom, Diskette 19,50 DM, Bestell-Nr. A 32.

**Dragon 32,** Blizzard, Kassette 8,– DM, Bestell-Nr. D 41

Space-Flight/Geosoft, Kassette 10,– DM, Bestell-Nr. D 51

Waregames, Kassette 8,50 DM, Bestell-Nr. D 61

Laser-Attack, Kassette 8,50 DM, Bestell-Nr. D 71

Hardcopy, Kassette 13,– DM, Bestell-Nr. D 81

Anwenderprogramm, Kassette 10,– DM, Bestell-Nr. D 91

Dragon Paint, Kassette 14,– DM, Bestell-Nr. D 101.

**ZX-Spectrum,** Inventur, Kassette 12,– DM, Bestell-Nr. S 41

Missile-Comment, Kassette 8,50 DM, Bestell-Nr. S 51

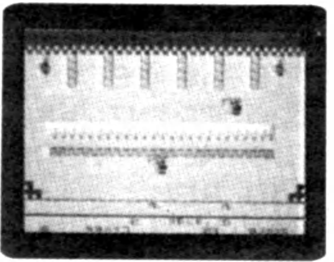
Defender/Lui der Wurm/Alternativer Zeichensatz, Kassette 13,50 DM, Bestell-Nr. S 61

Matheprogramm/Bongo-Beecatcher, Kassette 12,50 DM, Bestell-Nr. S 71

Solitaire/Superstat/Kleinstes gem. Vielfache, Kassette 14,50 DM, Bestell-Nr.

Jump about, Kassette 14,50 DM, Bestell-Nr. S 91

Pac-Man/Oil Panic, Kassette 14,50 DM, Bestell-Nr. S 101



Frogger, Kassette 16,– DM, Bestell-Nr. S 121

Jump, Kassette 14,50 DM, Bestell-Nr. S 22

Jet Set Freddie, Kassette 8,– DM, Bestell-Nr. S 32.

**ZX-81,** Go-Ball/Grand-Prix, Kassette 10,– DM, Bestell-Nr. Z 51

Moon-Crash/ZX-Draw, Diskette 10,– DM, Bestell-Nr. Z 61

Tonprogramm/Aldebaran, Kassette 10,– DM, Bestell-Nr. Z 71

Reversi, Kassette 10,– DM, Bestell-Nr.Z 91

Panik Labyrinth, Kassette 10,– DM, Bestell-Nr. Z 101

Expedition, Kassette 10,– DM, Bestell-Nr. Z 121

Spinnen, Kassette 14,50 DM, Bestell-Nr. Z 22

Spukhaus, Kassette 14,50 DM, Bestell-Nr. Z 32.

# CPU-SOFTWARE-SERVICE

**Commodore 64,** Multi-Key/S-Tool 64/Interrupt-Programme, Kassette 16,– DM, Diskette 21,– DM, Bestell-Nr. UC 2/85.

Spritehilfe/Diskloader/Directory, Diskette 21,– DM, Bestell-Nr. UC 3/85.

Tape-Directory/Asmon/Data-Generator, Kassette 15,– DM, Diskette 21,– DM, Bestell-Nr. UC 3/85.

Fast-Load, Kassette 10,– DM, Bestell-Nr. UC 4/85.

Diskmonitor/Zeichensatz, Diskette 20,– DM, Bestell-Nr. UC 5/85.

Reassembler/Maskengenerator, Kassette 15,– DM, Diskette 20,– DM, Bestell-Nr. UC 6/85.

Del 64/Treiberprogramm für 1526/MPS 802, Decelerator, Kassette 14,– DM, Diskette 19,– DM, Bestell-Nr. UC 7/85

**Schneider CPC-464,** Autorennen, Kassette 11,– DM, Bestell-Nr. US 3/85.

Universal-Datei V1, Kassette 9,– DM, Bestell-Nr. US 4/85.

Computerschrift/Symbol-Swap, Kassette 10,– DM, Bestell-Nr. US 5/85.

Keyboard Toolkit/Farbdekodierung, Kassette 11,– DM, Bestell-Nr. US 6/1/85.
Aladin, Kassette 9,– DM, Bestell-Nr. US 6/2/85.
Painter/ Box-Befehl, Kassette 14,– DM, Diskette 24,– DM, Bestell-Nr. US 7/85

**VC-20,** Sprites/Space-Battle, Kassette 13,– DM, Diskette 19,– DM, Bestell-Nr. UV 2/85.

Grafik-Painter, Kassette 10,– DM, Bestell-Nr. UV 5/85.

Vic-Clock, Kassette 9,– DM, Bestell-Nr. UV 6/85.

Decelerator, Kassette 9,– DM, Bestell-Nr. UV 7/85.

**Apple,** Asteroids, Diskette 18,– DM, Bestell-Nr. UA 2/85.

Dateiverwaltung, Diskette 18,– DM, Bestell-Nr. UA 3/85.

Digiter/Reset-Saver, Diskette 18,– DM, Bestell-Nr. UA 4/85.

Assambler-Monitor, Diskette 17,– DM, Bestell-Nr. UA 5/85.

Disk Check IC/Dos Change, Diskette 18,– DM, Bestell-Nr. UA 6/85.

Hello, Diskette 17,– DM, Bestell-Nr. UA 7/85.

**ZX-81,** The Castle of Questionmark, Kassette 10,– DM, Bestell-Nr. UX 2/1/85.

**TI-99,** Olympics, Kassette 11,– DM, Bestell-Nr. UT 2/2/85.

Funktionsanalyse/Hardcopy, Kassette 11,– DM, Bestell-Nr. UT 3/85.

Invaded by the Empire, Kassette 11,– DM, Bestell-Nr. UT 4/85.

Multicolor/Load-Programm, Kassette 14,– DM, Diskette 19,– DM, Bestell-Nr. UT 6/85.

Disassembler, Diskette 19,– DM, Bestell-Nr. UT 7/85.

**ZX-Spectrum,** Panik Sam/Colour Change/Clock, Kassette 13,– DM, Bestell-Nr. UZ 2/85.

MC-Routinen/Data-Base/Pixel, Kassette 11,– DM, Bestell-Nr. UZ 3/85.

Reversi/UDG-Editor, Kassette 15,– DM, Bestell-Nr. UZ 4/85.

Star-Basic, Kassette 13,– DM, Bestell-Nr. UZ 5/85.

Fill-Programm/Copy it, Kassette 14,– DM, Bestell-Nr. UZ 6/85.

Quassi, Kassette 13,– DM, Bestell-Nr. UZ 7/85.

**Atari,** Mauerflucht/Lunar-Simulation, Kassette 13,– DM, Diskette 19,– DM, Bestell-Nr. UI 2/85.

Ram-Tester/Cherry-Harry, Kassette 13,– DM, Diskette 19,– DM, Bestell-Nr. UI 3/85.

Crazy Egon, Kassette 13,– DM, Bestell-Nr. UI/85.

Monitorprogramm, Kassette 11,– DM, Bestell-Nr. UI 6/85.

Zeicheneditor, Diskette 19,– DM, Bestell-Nr. UI 7/85.

**MSX,** Zeichensatzkonverter, Kassette 10,– DM, Bestell-Nr. UM 3/85.

Zirkus, Kassette 13,– DM, Bestell-Nr. UM 5/85.

Pipeline, Kassette 13,– DM, Bestell-Nr. UM 4/85.

Alien Attack, Kassette 13,– DM, Bestell-Nr. UM 7/85.

**Bestellung per Telefon:**

Wenn es schnell gehen soll ... rufen Sie uns an. Wir nehmen Ihre Bestellung gern entgegen.


**Tel.-Nr. 0 56 51 / 4 06 93**
**oder 0 56 51 / 4 06 43**

**Tronic-Verlag**
**Postfach 41**
**3444 Wehretal 1**


## CPU Bestellkarte-Softwareservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service über den Verlag bezogen werden.
**(Ausland nur gegen Vorkasse)**

Die Zustellung erfolgt: gegen **Vorkasse** ☐ oder Inland per **Nachnahme** ☐ + Versandkosten

innerhalb von 1 Woche

Entnehmen Sie bitte aus unseren Preislisten die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir:

☐ Cassette für ______ (System) ☐ Anzahl

☐ Diskette für ______ (System) ☐ Anzahl

zum Preis von gesamt ______ DM

Name/Vorname: ______ Straße, Nr.: ______

PLZ/Ort: ______ Datum, Unterschrift ______

# „Decelerator 64/20“: UTILITY für den C-64 / VC-20

## Die Anwendungsmöglichkeiten von Decelerator 64/20:

Alle Computerfreaks, die schon einmal mit der Sprache Latein konfrontiert wurden, werden beim Lesen des Titels schon den Zweck des Programmes erkennen und mit froher Erwartung sofort den Commodore starten und das Programm eintippen. Für den Rest der Leser sei gesagt, daß man mit Decelerator die Exekution von jeglichen Betriebssystemroutinen, jeglichen Basicprogrammen bzw. Complicaten von z. B. Petspeed, Austrocompiler und vielen Maschinencodeprogrammen beliebig verlangsamen kann. Weiterhin kann man diese Programme an jedem beliebigen Punkt anhalten und später genau an dieser Stelle mit dem Programm fortfahren. Dies gilt auch für Programme, bei denen die RUN/STOP-Taste abgeschaltet ist.

So kann man endlich die sehr schnellen Arcadegames meistern oder der Versuch, den Ultimate-High-Score zu erlangen, scheitert nicht am Schellen des Telefons. Aber auch für andere Anwender bietet Decelerator neue Perspektiven, z. B. kann man mit der Hilfe von Decelerator die Programme endlich mit einer sinnvollen Geschwindigkeit auf dem Bildschirm listen, so daß die Fehler nicht wie sonst schon vor ihrer Entdeckung weggescrollt sind. Der hämische can't contiue Error ist vergessen, da das Programm nun immer wieder gestoppt und gestartet werden kann. Selbst für den Maschinensprachecrack bietet Decelerator ungeahnte Möglichkeiten, endlich kann er die einzelnen Stufen der viel bewunderten Sprite-Animation erkennen oder ohne langes Hacken den Frequenzgang der Spitzensoundeffekts erkennen.

Ein weiterer Pluspunkt von Decelerator ist die Compatibilität mit wichtigen Utilities wie Fastloadern, Turbo-Tape, Turbo II, Speed-Load. Diese Utilities laufen ohne Schwierigkeiten. Externe-Resets übersteht Decelerator auch, so daß ein ständiges Reloading entfällt.

**Wie funktioniert jetzt eigentlich Decelerator?**

Decelerator verlängert einfach die Bearbeitungszeit der Interrupts, indem er, bevor er zur Interrupt-Betriebssystemroutine springt, einfach eine Warteschleife durchläuft. Die Länge der Bearbeitungszeit der Schleife kann durch Druck der „+“- und der „–“-Taste verändert werden. „+“ für ein schnelleres Tempo und „–“ für ein langsameres. Die Funktion von Decelerator setzt also den Aufruf der Hardware-Interruptroutine voraus. Leider gibt es aber den Maschinensprachbefehl SEI (Set Interrupt disable status), der jegliche Interrupts abschaltet, somit auch Decelerator außer Gefecht setzt. Aber keine Angst, so leicht gibt sich Decelerator noch nicht geschlagen. Es ist nämlich ein Befehl zum Wiedereinschalten der Interrupts implementiert. Der einmalige Druck der Restore-Taste (bitte ohne Run/Stop) genügt. Decelerator macht sich hierbei die Non Maskable Interrupts zunutze, die bei C-64 / VC-20 z. B. durch die Restore-Taste ausgelöst werden können. Diese Interrupts sind softwaremäßig nicht ausschaltbar. Bei einem NMI springt der Prozessor zu einer Routine, die ihm durch die Vektoren $FFFA und $FFFB angegeben wird. Diese NMI-Routine muß also nur so verändert werden, daß sie das vor dem Interruptansprung auf den Stapel gesicherte Statusregister zurückholt, einen CLI (Clear Interrupt disable bit) durchführt und darauf das Statusregister auf den Stapel wieder zurücklegt.

Decelerator verfügt zuzüglich noch über eine extra Stop-Taste. Es handelt sich dabei um die „£“-Taste. Diese Taste veranlaßt, daß der Computer nicht mehr aus dem Interrupt zurückkommt, sondern eine Schleife durchläuft. Diese Schleife kann nur durch den NMI unterbrochen werden, der eine Flag setzt, die den normalen Interrupt veranlaßt, die Schleife zu verlassen. Zur Fortsetzung des Programms muß deshalb wieder die Restore-Taste gedrückt werden. Falls die Tastatur danach blockiert sein sollte, hilft das Motto: „2mal hilft besser. – Siehe da, alles wieder beim alten.“

**Die versionsspezifischen Merkmale von Decelerator:**

Es ist wohl jedem klar, daß wegen des unterschiedlichen Aufbaues von VC-20 und C-64 gewisse Veränderungen bei der Realisierung des Programmziels vorgenommen werden mußten. Der C-64 bietet z. B. die Möglichkeit, das Betriebssystem ins Ram zu verlegen, deshalb ist Decelerator 64 auch wesentlich complexer geworden und die C-64-Besitzer dürfen fast doppelt so viele der verhaßten Datas eingegeben. Ich war trotzdem bestrebt, das Programm so kurz wie möglich zu halten, um den Eingabefrust auf ein Minimum zu verkürzen. Deshalb sei auch gesagt, daß Rams weggelassen und Prints selbstverständlich verkürzt werden können. Decelerator verfügt über eine Data-Check-Funktion, die auf Data-Errors aufmerksam macht. Trotzdem ist es ratsam, das Programm vor dem RUN-Befehl abzusaven, da sonst evtl. das ganze Tippen umsonst war.

**Decelerator 64:**

Die Initialisierungsroutine kopiert zuerst das Rom ins Ram, schaltet dann das Rom aus und verändert das Betriebssystem so, daß die Interrupt- und die NMI-Routine von Decelerator mit ausgeführt werden und außerdem beim NMI nicht wie sonst zum Modulstart gesprungen wird. Decelerator kann nur durch den Poke 1,55, der wieder das Rom einschaltet, oder durch ein Überschreiben seiner Routinen ausgeschaltet werden. Man kann also davon ausgehen, daß die Programme, die mit Decelerator imcompatibel sind, zur Programmsicherung solche Befehle enthalten. Den potentiellen Hakkern unter den Lesern sollte es also nicht schwerfallen, zum Erlangen einer Compatibilität diese Befehle mit Hilfe des Diskmonitors durch NOPs zu ersetzen.

Nach einem erfolgreichen Reset kann man Decelerator 64 übrigens mit Poke 1,53 wieder einschalten.

Als letzter Tip sei gesagt, daß man beim Loading immer die normale Betriebssystemgeschwindigkeit verwenden sollte, da es sonst zu LOAD ERRORs kommen kann. Dies wird durch Druck der „+“-Taste erreicht, den man fortsetzen muß, bis der Cursor etwa wieder mit normaler Geschwindigkeit blinkt.

**Die Compatibilität von Decelerator 64:**

Zum Leid der C-64-Besitzer sei gesagt, daß Decelerator 20 eine weit höhere Compatibilitätsquote hat, als die C-64-Version. Dies erklärt sich daraus, daß die Professionals es inzwischen besser gelernt haben, ihre Programme zu sichern, und dazu auch sehr viel mehr Speicher auf dem C-64 zur Verfügung steht. Trotzdem läßt sich die Liste

der compatiblen Programme sehen. Sie ist bei weitem nicht vollständig, da nur ein Bruchteil der auf dem Markt befindlichen Programme getestet wurde. Es sei darauf hingewiesen, daß einige Programme der Liste möglicherweise bei verschiedenen Lesern incompatibel sind, da die Firmen ihre Programme manchmal im Nachhinein mit verbesserten Sicherungen dupliziert haben.

**Compatibel sind:**
Hunchback, Scramble, B.C.'s Quest for Tires, Attack of the Mutant Camels, Matrix, Vortex Raider, Gridrunner, Laser-Strike, Mangrove, Triad 64, Puckman, Pucman, Pedestrian, Hyper-Hen, Donkey Kong II, Pacacuda, Hexpert, Crossfire, Gallaga, Trax, Helicommand, Burger-Crash, Jet Pac, Chaser etc.

```
10 REM   ************************
11 REM   *                      *
12 REM   *    DECELERATOR 64    *
13 REM   *                      *
15 REM   ************************
16 REM   *                      *
17 REM   *       (C) 1985       *
18 REM   *                      *
19 REM   *          BY          *
20 REM   *                      *
25 REM   *    EBERHARD  AUST    *
27 REM   *                      *
30 REM   ************************
70 PRINT"■----------------------------------------";
80 PRINT"   DECELERATOR 64 (C) 1984 BY E. AUST   "
85 PRINT"----------------------------------------";
90 REM
91 REM   DATA EINLESESCHLEIFE 1
92 REM
100 FORA=0TO101:READB:PRF=PRF+B:POKEA+6000,B:NEXT
150 IFPRF<>13349THENPRINT"■DATA ERROR IN VORBEREITUNGSROUTINE":END
190 REM
191 REM   DATA EINLESESCHLEIFE 2
192 REM
200 FORA=0TO122:READB:ARF=ARF+B:POKEA+53063,B:NEXT
250 IFARF<>16055THENPRINT"■DATA ERROR IN HAUPTROUTINE":END
300 PRINT"■■                   DATAS OKAY■■"
305 REM
306 REM   STARTEN DER VORBER.ROUTINE
307 REM
310 SYS6000
315 PRINT"────────────────────────────────────────";
317 PRINT
320 PRINT"SCHALTEN SIE NUN DECELERATOR DURCH DRUCKVON RUN/STOP UND RESTORE EIN"
323 PRINT
325 PRINT"────────────────────────────────────────";
330 GOTO330
1000 REM
1001 REM   VORBEREITUNGSROUTINE
1002 REM
1010 DATA160,160,162,0,140,124,23,140,127,23,189,0,191,157,0,191,232,224,0
1020 DATA240,3,76,122,23,200,192,192,240,3,76,116,23,160,224,162,0,140,156,23
1030 DATA140,159,23,189,0,225,157,0,225,232,224,0,240,3,76,154,23
1040 DATA200,192,0,240,3,76,148,23,169,229,141,214,253,169,76,141,68,254,169
1050 DATA71,141,69,254,169,207,141,70,254,169,234,141,91,254,141,92,254,141
1060 DATA93,254,120,169,53,133,1,88,96
2000 REM
2001 REM   HAUPTROUTINE
2002 REM
2010 DATA169,46,133,255,40,88,8,120,169,76,141,49,234,169,53,141,1,0,169
2020 DATA108,141,50,234,169,207,141,51,234,76,71,254,0,0,0,0,0,0
2030 DATA169,255,141,2,220,165,197,201,40,208,10,174,160,207,224,0,240,3
2040 DATA206,160,207,201,43,208,10,174,160,207,224,255,240,3,238,160,207
2050 DATA201,48,208,10,165,255,201,46,240,3,76,147,207,234,162,0,224,0,208
2060 DATA3,76,184,207,160,0,200,192,40,240,3,76,168,207,232,76,159,207
2070 DATA0,0,0,234,169,0,133,255,32,234,255,76,52,234
3000 REM
3010 REM   ENDLICH GESCHAFFT !!!!
3020 REM
```

**Decelerator 20:**
Decelerator 20 läuft auf dem VC 20 mit einer 8K- oder einer 16K-Ram-Extension. Er bietet die Möglichkeit, Programme, die für die Grundversion (3.5K Ram) oder für den mit 8K Ram erweiterten Computer geschrieben sind, auszuführen. Die Grundversion wird initialisiert, wenn die Frage, ob die Speichererweiterung ausgeschaltet werden soll, mit „J" beantwortet wird.
Da eine Modifikation des Betriebssystems auf dem VC-20 wegen des Ram-Mangels nicht möglich ist, werden die Routinen von Decelerator durch Veränderung des Hardware-Interrupt-Vektors ($314–$315) und des NMI-Vektors ($318–$319) aufgerufen. Diese werden allerdings nach dem Druck von RUN/STOP + RESTORE wieder auf ihre normalen Werte gesetzt, so daß Decelerator durch Eingabe von SYS 15310 neu initialisiert werden muß. Auch nach externen Resets kann Decelerator durch diesen SYS neu gestartet werden.
Bitte beachten Sie auch bei Decelerator 20, daß beim LOADING möglichst Normaltempo gewählt werden sollte.

**Die Compatibilität von Decelerator 20:**
Hier verliefen alle 3.5K-Ram-Tests positiv. Getestet wurden z. B. Mad-Painter, Frogger, Earth Defense, Hell Driver, Exterminator, Scramble, Fire, Bounce-Out, Pacer, Pit, Blockade etc. Diese Liste könnte beliebig fortgesetzt werden.
Die meisten Programme mit 8K-Ram-Extension laufen ebenfalls, nur solche, die absolut das ganze Ram füllen, überschreiben die Routinen, von Decelerator. Für 16-Ram-Extension-only-Programme kann eine Compatibilität durch Zurückversetzen von Decelerator um 8K erreicht werden. Für jeden, der Maschinensprachkenntnisse besitzt, sollte dies kein Problem sein, da nur die Adressen um 32 Pages erhöht werden müssen.

```
10 REM ***************
15 REM *             *
20 REM *DECELERATOR20*
22 REM *             *
25 REM *  (C)  1985  *
27 REM *             *
30 REM *     BY:     *
33 REM *             *
35 REM *EBERHARD AUST*
37 REM *             *
40 REM ***************
60 PRINT"▒----------------------     DECELERATOR 20▒"
70 PRINT"   (C) 1985  E.AUST    ----------------------▒▒▒▒▒▒▒"
100 REM
101 REM EINLESESCHLEIFE
102 REM
110 FORA=0TO111:READB:POKE15300+A,B:PRF=PRF+B:NEXT
140 REM
141 REM DATAS PRUEFEN
142 REM
150 IFPRF<>12132THENPRINT"▒DATA":PRINT"ERROR":END
180 REM
181 REM DECELR. STARTEN
182 REM
190 SYS15310
200 PRINT"----------------------       DATAS O.K.   ▒ "
210 PRINT"DECELERATOR IN ACTION"
220 PRINT"----------------------▒"
222 REM
223 REM 8/16 K RAMEXT.
224 REM  AUSSCHALTEN
225 REM
230 PRINT"RAMEXT. AUS (J) ";
240 INPUTA$
250 IFA$="J"THENPOKE641,0:POKE642,16:POKE643,0:POKE644,30:POKE648,30:SYS64824
1000 REM
1001 REM NMI ROUTINE +
1002 REM VORBRTS.ROUT.
1003 REM
1010 DATA169,46,133,255,40,88,8,76,173,254,120,169,241,141,20,3,169,59,141,21,3,
169
1020 DATA196,141,24,3,169,59,141,25,3,96
2000 REM
2001 REM HAUPTROUTINE
2002 REM
2010 DATA169,0,133,255,76,191,234,0,0,0,0,0,165,197,201,5,208,10,174,32,60,224
,0
2020 DATA240,3,206,32,60,201,61,208,10,174,32,60,224,255,240,3,238,32,60,201,6
2030 DATA208,10,165,255,201,46,240,3,76,19,60,234,162,0,224,0,208,3,76,228,59,16
0,0
2040 DATA200,192,50,240,3,76,40,60,232,76,31,60
2050 REM
2060 REM GESCHAFFT !!!
2070 REM
```

# LaserWriter von Apple

LaserWriter heißt das neue Flaggschiff unter den Apple-Peripherie-Geräten, das sich mit seinen multifunktionalen Leistungsmerkmalen deutlich aus der allgemeinen Druckerwelt hervorhebt. Eingebaute Zeichensätze geben dem Anwender den Vorteil, Zeichen und Grafik in Laser-Auflösung zu erhalten und zu mischen. Die Qualität kommt der von Satzmaschinen nahe und ergibt sich aus der ungewöhnlich hohen Auflösung von 300 Punkte/Inch gegenüber 70 Punkte/Inch bei „Normaldruckern.

Verschiedene Schrifttypen bieten die uneingeschränkte Möglichkeit, Formulare, Berichte, Geschäftsgrafiken und sogar Präsentationsfolien selbst zu erstellen.

Erreicht werden diese vielfältigen Leistungsmerkmale durch einen eingebauten Mikrocomputer mit 68000-Prozessor, 1,5 MB RAM und 0,5 MB ROM.

Ferner benutzt der LaserWriter PostScript, einen neuen Standard für hochauflösende Ausgabegeräte. Damit ist der LaserWriter kompatibel zu Satzmaschinen wie z. B. Linotype.

Kostengünstig wirkt sich die eingebaute Apple-Talk-Schnittstelle aus. Über AppleTalk können 31 Macintosh zusammen den LaserWriter benutzen.

Auf dem LaserWriter können auch IBM-PC und Kompatible drucken. Dieser Drucker besitzt eine eingebaute Diablo 630-Emulation. Ein PC kann über eine RS 232-Schnittstelle direkt an den LaserWriter angeschlossen werden. Der LaserWriter wird dann statt einem Diablo-Typenraddrucker verwendet. Diablo 630 ist eines der meist benutzten Protokolle. Eine weitere Anschlußmöglichkeit ist über die PC-Karte von Apple gegeben. Damit kann der PC über AppleTalk auf dem LaserWriter drucken. Dabei wird wieder über das im LaserWriter eingebaute Diablo 630-Protokoll gedruckt. Die limitierten Grafikmöglichkeiten des IBM-PCs sind damit allerdings nicht auszugleichen.

Die Apple-Macintosh-Software ist mit dem LaserWriter kompatibel, so daß z. B. professionelle Softwarepakete wie JAZZ von Lotus oder die Software-Serien von Microsoft genutzt werden können.

## Der LaserWriter auf einen Blick

| | |
|---|---|
| Eingebaute Schrifttypen: | Helvetica<br>Times<br>Courier<br>Symbols |
| Laserfähige Software: | Sämtliche Apple Macintosh-Software, Lotus JAZZ. Sämtliche Microsoft-Produkte für Macintosh sowie neue Spezialprodukte wie: Page Maker von Aldus Corp für den Satz und die Komposition von Mitteilungsblättern, Prospekte, Datenblätter. Nahezu alle anderen Macintosh-Produkte können ohne Modifikation eingesetzt werden. |
| Grafik-Sprache: | PostScript von Aldobe Systems Inc.. PostScript ist systemunabhängig, so daß auch ältere PC-Standard über LaserWriter drucken können. |
| Netzwerkfähigkeit: | AppleTalk bereits eingebaut. |
| Prozessor: | Motorola 68000 (12 MHz) mit 0,5 MB ROM und 1,5 MB RAM. |
| Druckwerk: | Canon LBP-CX 10. |
| Ausgabemedium: | Papier, OH-Folie. |
| Auflösung: | 300 Punkte pro Inch. |
| Wartung: | Durch autorisierte Apple-Händler mit Macintosh Office-Spezialausbildung. Der Toner-Austausch kann vom Anwender selbst vorgenommen werden. Eine Kartusche für 2000–3000 Seiten. |
| Garantie: | 12 Monate. |
| Lieferbar: | Seit Juni 1985. |
| Preis: | DM 27 500,– (unverbindliche Preisempfehlung inklusiv 14 % MwSt.). |

# Sonderpreis! CPU-Kleinanzeigen jetzt zum Sonderpreis!

**Verkaufe Originalprogramme** oder tausche gegen andere Originale ein! Deus ex machina 33, Tir na nog 22, Valhalla 22, Pimania 15, Jet Set Willy 12, King Arthur's Quest 19, PI-Eyed 9, PI-Inre 9, Piromania 9, New Wheels John? 7, GOTO JAIL 6, PI-Balled 6, Backgammon 6, Psion Chess 6, Chess (Artic) 9, Flight Simulation 6, + 1 DM für Porto u. Verpackung. Georg Scheibe, Hebeckenkamp 9, 4355 Waltrop, Tel. 0 23 09 / 7 46 19

**Verkaufe** Atari-Recorder, Tel. 0 21 52 / 51 02 43

**Verk.** TI99/4A + Ex. Basic + Cass.R + Kabel + Mod. Espial + J.-Adapter + 55 Progr. auf 4 Cass. DM 730,–, auch einzeln. Tel. 07 11 / 53 96 12

**Verkaufe TI-Soft-/Hardware u. Bücher!!** Es lohnt sich! Infos gegen -,60 DM in Briefmarken bei: Ralf Derissen, Josef-Lambertz-Str. 21, 5120 Herzogenrath

**NDR-Klein-Computer,** 10 Platinen, funktioniert, aus Zeitmangel zu verk., Neupreis inkl. Monitor 1800,–, VB 900,–. R. Niedermeier, Tel. 0 80 67 / 6 95

**Wegen Hobbyaufgabe** abzugeben 30 Lehr- und Übungsprogramme auf Cassette in Basicode-2 für nur DM 85,– (Vorkasse). Johann Feddermann, Postfach 10 21 02, 2000 Hamburg 1

**Massenweise Niedrigpreise!** Für C-64, VC-20, C-16, ATARI, SPECTRUM, SCHNEIDER + ZAXXON, BLUE MAX, GHOSTBUST. ab 35,–; FL. SIM. II 136,–; Info für 1,50 (in BM) von H. Topf, Softw. & Serv., Alfred-Bucherer-Str. 63, 5300 Bonn 1, Tel. 02 28 / 62 13 92

**Anwenderprogramme** für die Top-Computer Schneider CPC und ZX-Spectrum. Gratisinfo anfordern bei Friedrich Neuper, 8473 Pfreimd, Postfach 72. Bitte Computertyp angeben!

**Tausche VC-64** Spitzenprogramme. Tauschlisten an: Jürgen Hödl, Muggenthaler Str. 10, 8391 Tittling

**TI ★ Rex Soft ★ Superprogramme! ★** Wir bieten ständig neue Programme zu echten Preishits!!! Großes Programmangebot!!! Info 1,– DM bei: Daniel Peier, Hulfteggstr. 31, CH-8400 Winterthur

**TI99/4A Ext.-Spiel** nur für Erwachsene, Alter ang., zum Preishit von nur DM 10,– inkl. Porto. Kass. u. Infoprogr. in Umschlag an T. Karbach, Remscheider Str. 18, 5650 Solingen 1

**TI99/4A Ext.** Jetzt können Sie Ihren Taschenrechner vergessen. Komfort. menügesteuertes Progr. mit 14 Rechenarten nur DM 10,–. In Umschlag an D. Karbach, Remscheider Str. 18, 5650 Solingen 1

**● ● ● TI99/4A ● ● ●** Drucke Ihre Listings, egal wie lang, auf Seikosha GP 100 A. Kass. + DM 10,– in Umschlag an D. Karbach, Remscheider Str. 18, 5650 Solingen 1

**TI99/4A:** Extended-Basic, P-Box, Disk-Drive, Disk-Controller, Console, 32k-Ram, kpl. VB 1800,– DM. Außerdem: Datenverw./Analyse, Text- u. Dateiverw., TI-Writer. Helmut Zindel, Tel. 0 41 94 / 75 64

**TI99/4A** Hardware u. Soft. Suche Peripherie-Box. A. Mlodoch, Postf. 1527, 4352 Herten

**TI 99:** Alpiner/Car Wars/ Tombstone City/Ti-Invaders/ Joysticks gegen Gebot. T.: 09 11 / 51 27 93

**TI99/4A:** TI und Ex-Basic-Programme zu verk., Info g. Rückporto, B. Knedel, Tulpengasse 16, 3171 Weyhausen, Tel. 0 53 62 / 7 11 87

**ZX 81 + 16 KRAM + Netzteil,** 5 Bücher + 6 Original-Spiele + 4 Softwarecassetten für DM 240,– abzugeben. Andreas Ipolt, Landeckstr. 3, 6748 Bergzabern, Tel. 0 63 43 – 43 48

**Suche** käuflich Zeitschrift Homecomputer 3/83 und 1-7/84, außerdem CPU 1-6/84 + 8/84, Listings müssen lesbar sein.

**Suche** Peri-Box für TI, möglichst voll ausgestattet. Haken hierbei: zahle gut, aber in Raten! Chiffre: TI-B

**Suche C 64,** gut erhalten, zahle bis 400,– DM oder Tausch gegen VC-20 + 6 KK + Superprogramme. A. Dinsch, Berliner Str. 17, 5980 Werdohl

**Spectrum.** Unser Club Wuppertal nimmt wieder Mitglieder auf! Neues Konzept! Für alle ZX-Freaks ein muß! Info gg. Rückporto. R. Knorre, Postf. 20 01 02, 5600 Wuppertal 2

**ZX-Spectrum.** Software-Tausch, schickt Eure Liste an Anton Meisinger, Ungargasse 11, A-7350 Oberpullendorf, Austria

**Statistik und System-Lotto für Spectrum 48K.** Viele statistische Werte u. autom. Übernahme ins System. 26(!) Auswahlkriterien. Mutlu, Eichenstr. 4, 6790 Landstuhl

**ZX-Spectrum:** Tape-Kopierer, kopiert fast alle Programme am Markt. Zeigt auch Header-Daten. 15,– DM / 99,– öS, W. Hübl, Klitschg. 18, A-1130 Wien

**★ Spectrum-Programme-Tausch ★** Über 100 versch. Programme. Kein Rückporto erf. Liste an: Detlef Thee, Schwartauer Allee 35, 2400 Lübeck 1, Tel. 04 51 / 47 49 30 ab 19 Uhr

**ZX Spektrum 48K, ZX Printer,** Software, Joystick zu verk., Preis 500,– DM komplett. J. Schneider, Flurstr. 7, 6581 Dickesbach, Tel. 0 67 84 / 89 42

**VC 20.** Suche 40/80 Karte und schaltbare Modulbox (billig!). Suche gute Programme (aller Art). Angebote **nur** schriftlich an Bernhard Helle, Schützenstr. 11, 4100 Duisburg 14

**Hallo VC-20-Freaks!** Wer hat Interesse in einem Club mitzumachen? Der CVC hat 13 Mitglieder und neue sind herzlich willkommen! Außerdem gibt's eine Clubzeitschrift. CVC c/o Dirk Weißenborn, Falkenweg 66, 7400 Tübingen

**Neu – das Mallbox-Telefonbuch-Rufnummer-Verzeichnis der DFÜ-Anbieter** einschl. Datex. p. mit monatl. Nachtrag (1 Jahr) zu beziehen für 8,50 DM. Tel. (04 21) 56 38 00 oder 56 38 31

**CPC464 ! ! !** Verlängerungs-Kabel zwischen Monitor und Keyboard mit passenden Steckern nur DM 20,– an Postscheck Hamburg 32572-201 oder Schein an H. Behrendt, Marrensberg 2, 2390 Flensburg

**CPC 464 Printer ›NLQ401‹** zu verk. VB 650,– DM, neu, 2 Mon., und einm. Originalprg., wie Rechnungssch., Text., Chess u. a., wegen Aufgabe abzugeben. Tel. 02 11 / 27 47 51 ab 16 Uhr

**CPC 464!! Super-Software:** Z. B. 3-D-Fußball oder Night Flight nur je DM 20,–!! Textverarbeitung etc. alles vom COMPI-Club, J. Heise, A. d. Linde 8, 5226 Reichshof. Info gratis!

**CPC464 ! ! !** 12 gute Spielprogramme auf Kassette, keine Raubkopien. DM 20,– an Postscheck Hamburg 404576-207 oder Schein im Brief an Hans-J. Behrendt, Neustadt 3, 3550 Marburg

Der 64'er SOFTWARE-CLUB aus Bochum sucht Verbindungen zu anderen Computer-Clubs.
**Club-Aktivitäten:** Club-Treffen, Club-Fahrten, Club-Zeitung „CLUB-NEWS" und Austausch von Programmen.
**Unsere Adresse:**
64'er SOFTWARE-CLUB
Oliver Nürnberg,
Holtingstraße 16c
4630 Bochum 1

Wir suchen Euch!?!
Alle Computerfreaks die bereit sind, folgende Dienste **kostenlos** in Anspruch zu nehmen bzw. zu teilen.
Wir bieten:
1. Softwaretausch
2. Entwicklung von Hard- u. Software
3. Beratung bei Problemen
4. Reparaturen
5. Programmierkurse
6. Erstellung einer Clubzeitung

Willkommen sind alle Systeme/ Computertypen.
Interessenten wenden sich bitte, wenn möglich bitte mit Rückporto, an den: IC Computerclub, Beethovenstr. 66, 4815 Schloß Holte-Stuk., Tel.: 0 52 07 / 8 77 00. Jeder Brief bzw. Anruf wird beantwortet.

## Kleinanzeigen

Liebe Leser,

wollen Sie eine Kleinanzeige aufgeben? Suchen Sie einen gebrauchten Home-Computer, Software, Hardware, Kontakte zu Computer-Clubs etc.?

**Dann nutzen Sie jetzt den Kleinanzeigenservice der „CPU": Die „CPU" ist ein kompetentes Anwender-Magazin.**

Sie erreichen durch uns eine Vielzahl von Lesern im In- und Ausland.

**Und so wird's gemacht:**

Kleinanzeigen-Coupon ausfüllen (alle Zeilen können ausgefüllt werden) und **einsenden an**

Tronic-Verlag
Postfach 41
3444 Wehretal 1

**Also,** schnell ausfüllen und einwerfen in den nächsten Postkasten. Dann ist mit Sicherheit Ihre Kleinanzeige dabei.

– CPU – Basar

## Kleinanzeigen

– CPU – Basar

Bitte veröffentlichen Sie ab nächstmöglicher Ausgabe unter der Rubrik:

☐ SOFTWARE ☐ HARDWARE ☐ VERSCHIEDENES

Mein Computersystem: ______

folgenden Text:

Bitte jeweils 28 Buchstaben pro Zeile! Bitte Absender nicht vergessen!

Preis für Kleinanzeigen: 10 DM je Kleinanzeige
(keine gewerbliche Nutzung) ☐ per Vorkasse ☐ Scheck ☐ Bar

Name/Vorname:

Straße, Nr.: PLZ/Ort:

Datum, Unterschrift

Ich versichere hiermit, daß die angebotenen Programme/Beiträge frei von Rechten Dritter sind!

# Neue Matrix-Drucker von OKIDATA

Die neuen MICROLINES ML 192 und ML 193 bieten eine hohe, wegoptimierte Druckgeschwindigkeit von 160 Zeichen/Sek. im Entwurfsdruck-Modus (für den schnellen Ausdruck von EDV-Listen) und ein gestochen scharfes Schriftbild im Schönschrift-Modus (für Dokumente und Geschäftsbriefe) mit 33 Zeichen/Sek. (automatischer Einzelblatteinzug als Zubehör).

Der ML 192 bzw. 193 verfügt über viele sinnvolle Sonderfunktionen, z. B. Dehnschrift, Hoch-/Tiefstellungen, horizontalen und vertikalen Fettdruck, kontinuierliches Unterstreichen, echte Unterlängen, 12 nationale Zeichensätze (inkl. ASCII (Ø), ASCII (0)), IBM-Zeichensätze, punktadressierbare Vollgrafik (bis 288 x 144 Punkte/-Zoll).

Die MICROLINES 192 und 193 können verschiedenste Papierarten verarbeiten, wobei der ML 193 durch seine größere Wagenbreite an zusätzlicher Flexibilität gewinnt (maximale Papierbreite beträgt 406,4 mm).

Eine Centronics-parallele Schnittstelle gehört zu der Standardausrüstung des neuen ML 192 bzw. ML 193.

## Die Gewinner unseres Preisrätsels aus dem Monat Mai!

**1. Preis:**
10 Kassetten aus unserem Software-Service
Michael Nahm, Ochsenfort

**2. Preis:**
3 Kassetten aus unserem Software-Service
Andreas Streng, Coburg

**3. Preis:**
2 Kassetten aus unserem Software-Service
Klaus Paradies, Rheine

**4.–30. Preis:**
H.-G. Hasebrock, Schüttorf
Bernd Michaelsen, Hamburg
Michael Burgmaier, Ertingen
Joachim Böhm, Georgensmünd
E. Vornberger, Randersacker
Jürgen Gerkens, Hamburg
Roland Hammelmann, Würzburg
Stefan Sieloff, Pullach
Günter Schmidt, München
Berthold Prause, Wendeburg
Gunnar von der Beck, Dillenburg
Alois Leimüller, Köstendorf
Alfred Rehberger, Feldheim
Wilfried Ganschinietz, Tierhaupten
Friedrich Tanzer, Wien (Austria)
Michael Gilles, Singen (Hohentwiel)
Michael Kircher, Bergheim
Reiner Weidmann, Nürnberg
Uwe Velker, Rheine
Reiner Dold, Filderstadt 1
Markus Eisenmann, Winnenden
Matthias Marx, Langenhagen
Karl Müller, Memmingen
Burkhard Blümke, Kempen 1
Gilbert Hönig, Overath
Paul Dietsch, Köln
Jens Schmidt, Bremen

# Korrektur zur CPU-Ausgabe Nr. 6 „Disk Check“ für den Apple-Computer

**Aufgrund technischer Probleme mit unserem Drucker sind beim Ausdrucken des Listings leider einige Fehler aufgetreten. Hier die Korrektur:**

```
]

]LIST 220

220  IF  LEN (T$) = 0 OR  VAL ( LEFT$
     (T$,1)) = 0 THEN L = 34

]LIST 300

300  NORMAL : HTAB 1: VTAB 18: PRINT
     "1  PRINT BAD SECTORS": PRINT
     : PRINT " RESTART": PRINT
     : PRINT "3  EXIT": PRINT : GET
     T$:X =  ABS ( INT ( VAL (T$)
     )): IF X > 3 OR X = 0 THEN 3
     00
```

```
]LIST 330

330  IF ERR = 0 THEN  HOME : HTAB
     15: VTAB 19: INVERSE : PRINT
     "NO ERRORS !": WAIT  - 16384
     ,128: HOME : GOTO 300

]LIST 350

350  VTAB 19: INVERSE : INPUT "OU
     TPUT-SLOT   (0-7  <DEFAULT 0
     > ) ? ";S$:S =  ABS ( INT ( VAL
     (S$))): IF S > 7 THEN 350

]

        8018-  00 03 8D 01 03 85 FF 8D
```

*Wenn Sie in der Lage sind, auch andere an Ihren Ideen teilhaben zu lassen, dann setzen Sie sich mit dem TRONIC-Verlag in Verbindung!*

*Wir suchen ständig neue Beiträge, z. B.:*

- **Tips und Tricks für die verschiedenen Computersysteme**
- **Kleine Maschinen-Routinen (Assembler-Listing)**
- **BASIC-Unterprogramme**
- **Hardware-Bauanleitungen**
- **und andere Beiträge**

*Wir brauchen von Ihnen:*

1. **Ausführliche Beschreibung des eingesandten Beitrages**
2. **Datenträger (Kassette oder Diskette)**
3. **Komplettes Listing (wenn zur Beschreibung erforderlich)**

All dies senden Sie bitte an den TRONIC-Verlag, Postfach, 3444 Wehretal 1

**Wir zahlen: 120,– DM pro abgedruckte Seite.**

# CPU-Clubvorstellung: SVI/MSC-Computer-Club

## Zielgruppe
Zielgruppe des Clubs sind alle Besitzer(innen) von SVI-Computern (SVI-318, SVI-328, SVI-728; bei entsprechender Nachfrage auch der Geräte Bondwell 12/14/16).

## Projekte
1) Herausgabe einer regelmäßig erscheinenden Club-Zeitschrift. Geplant ist ein vierteljährlicher Rhythmus.
2) Erfahrungs-Austausch: Hierzu gehören Soft- und Hardware-Probleme und deren Lösung, Neuentwicklungen, günstige Einkaufsquellen usw.
3) Software-Börse: Hier wünschen wir uns einen regen Austausch von selbstgeschriebenen Programmen. Bevorzugte Programmiersprachen sind (neben BASIC) PASCAL und „C", außerdem Z80-Maschinensprache unter CP/M und BASIC.
4) Kontakte zu anderen User-Clubs: Obwohl wir uns als „offizieller" User-Club verstehen (mehr dazu später), sind wir daran interessiert, mit evtl. schon bestehenden Clubs Kontakt aufzunehmen. Dazu gehören auch – entsprechende Nachfrage vorausgesetzt – die MSX-Clubs anderer Hersteller.

## Mitgliedsbeitrag
Da wir wissen, daß viele SVI-Besitzer noch Schüler sind, wollen wir einen möglichst niedrig gehaltenen und außerdem gestaffelten Mitgliedsbeitrag einführen. Dieser Beitrag soll in erster Linie dazu dienen, die Unkosten für die Herausgabe der Club-Zeitschrift zu decken; geplant ist ein Jahresbeitrag von DM 30,– für Schüler und DM 60,– für den „Rest der Welt". Wir vertrauen hier auf die Ehrlichkeit unserer Mitglieder und bitten ausdrücklich darum, daß sich nicht alle Besserverdienenden unter dem Namen ihrer Söhne und Töchter anmelden!

## Die augenblickliche Lage
Bisher können wir folgende Leistungen und Kontakte vorweisen:
1) Unterstützung durch die deutsche SVI-Vertretung und Anerkennung als „offizieller" User-Club.
2) Kontakte zum SVI-Club Österreich.
3) Vermittlung besonders gelungener Programme und Hardware-Zusätze an ein professionelles Systemhaus.
4) Telefonischer Feuerwehrdienst für akute Soft- und Hardware-Schwierigkeiten unter den Nummern (02 09) 20 52 42 (Uwe Schröder) für den norddeutschen und (0 61 52) 5 84 52 (Rainer Monzheimer) für den süddeutschen Raum.

Wir würden uns freuen, Sie als Mitglied begrüßen zu können. Falls Sie sich zu einem Beitritt entschließen, teilen Sie uns dies bitte mit, unter Angabe der folgenden Punkte:
1 Welche Geräte (Grundgerät, Expander, Diskettenlaufwerk(e), Drucker usw.) besitzen Sie?
2) Welche Software können Sie evtl. anbieten, woran sind Sie besonders interessiert?
3) Welche Erfahrungen (positiv oder negativ) haben Sie bisher mit SVI gemacht?
4) Was erwarten Sie von einem SVI-Club?

Anschrift:
Uwe Schröder
4650 Gelsenkirchen
Wannerstraße 57

# Club-Vorstellung: Rex Soft-TI-User-Club

Unser Club besteht noch nicht sehr lange ...
Zuerst war Rex Soft „nur" eine kleine Gruppe von TI-Usern, die Software herstellen, doch als wir feststellten, wie stark das Interesse am TI ist, haben wir uns entschlossen, daraus einen Club zu machen ...
Der Club besteht zur Zeit aus über 30 Mitgliedern, aber Rex Soft ist stark am wachsen ...
Uns scheint es, daß es noch viel mehr TI-User gibt, die Interesse an einem solchen Club haben, darum möchten wir unseren Club ein wenig vorstellen.
Unser Club versucht alle Probleme zu lösen ...
Wenn ein Mitglied etwas auf dem TI programmieren will, und er weiß nicht wie, dann kann er sich an die anderen Rex Soft Mitglieder wenden.
Da unsere Mitglieder aus verschiedenen Ländern kommen, haben wir uns für brieflichen oder telefonischen Kontakt entschieden.
Die Zentrale von Rex Soft ist in der Schweiz, aber uns scheint es sehr wichtig, daß sich die Mitglieder untereinander auch kennen. Wir haben auch eine eigene Clubzeitschrift, die etwa monatlich erscheinen wird.
Diese Zeitschrift haben wir noch als Programm gestaltet, um diese für alle Mitglieder gratis anbieten zu können ...
Unsere Zeitschrift soll von allen Mitgliedern zusammen gestaltet werden. Hier werden Probleme behandelt, die für alle wichtig sind. Weiter gibt es Tips & Tricks, die noch (fast) niemand kennt, außerdem existiert eine Ideenecke, und Softwaretests von Programmen, die Rex Soft Mitglieder durchgeführt haben.
Ebenfalls möchten wir in dieser Zeitschrift eine Top Ten über Programme, die in TI-Basic oder X-Basic geschrieben sind, bringen. Bei Rex Soft ist noch viel mehr geplant ... Wir suchen auch ständig nach neuen Ideen ...
Bei uns ist was los ...!
Wenn Ihr das Info wollt, legt Ihr am besten 1 DM Rückporto bei ...
Anschrift: Rex Soft, Daniel Peier, Hulfteggstraße 31, CH-8400 Winterthur

**Nutzen Sie die Möglichkeit zur kostenlosen Vorstellung Ihres Computer-Clubs in der CPU!**

# Der Sharp 16-Bit-Bürocomputer aus der MZ-5600-Serie

Dieser 16-Bit-Computer besitzt eine Speicherkapazität von 256 KB Standard und ist erweiterbar auf 512 KB. Das Diskettenlaufwerk schafft Speicherplatz für 800 KB pro Laufwerk und das Modell MZ-5645 ist standardmäßig sogar mit einem integrierten 10-MB-Festplattenlaufwerk ausgestattet.

Bemerkenswert ist vor allem der große Video-RAM-Speicher von zusätzlich 96 KB mit einem Anzeige-Kontroll-LSI für exzellente Grafikdarstellungen: Die Anzeige erfolgt im Video-RAM bitweise, d. h., daß ein Punkt einem Bit entspricht, um ein sauberes, klares Bild zu gewährleisten. Dies ist auch beim „Rollen" vorteilhaft. Durch „punktweises Rollen" wird die Lesbarkeit der Zeichen erhalten, auch wenn sich die Anzeige verschiebt. Für die Arbeit mit Farbgrafiken sind die Farbauswahlfunktion und die Farbprioritätsfunktion wertvoll.

Der zusätzliche Speicherplatz im Video-RAM schafft ebenfalls die Möglichkeit der Window-Darstellung. Bis zu vier verschiedene Bildschirmmasken können als Fenster auf dem Bildschirm gleichzeitig angezeigt werden. Für die schnelle Verarbeitung der Daten und Grafiken sorgt der 8086-Mikroprozessor mit 16-Bit-CPU. Das Betriebssystem EOS 16 ist anwendungskompatibel zu CP/M-80, CP/M-86, CCP/M-86 und MS-DOS. Damit kann die MZ-5600-Serie auf einen enormen Software-Markt vom Start weg zurückgreifen – übrigens auch auf das sehr weit verbreitete Angebot der 8-Bit-Software, die mit dem angebotenen Z-80-Board sofort lauffähig ist.

Die Multiskating-Funktion ermöglicht einen gleichzeitigen parallelen Programmablauf.

Als Optionen bietet Sharp eine „Mouse" als Eingabeeinheit, die komplizierte Eintastfolgen erübrigt. Außerdem gibt es den numerischen Datenprozessor 8087 für besonders schnelle Berechnungen. Als Bildschirm ist wahlweise ein 12"-Monochrom oder 15"-Farbbildschirm erhältlich.



Für den umfangreichen Ausbau der Peripherie bietet das System eine Vielzahl eingebauter Schnittstellen, u. a. Centronics, 2 Kanal RS-232C synchron und asynchron.

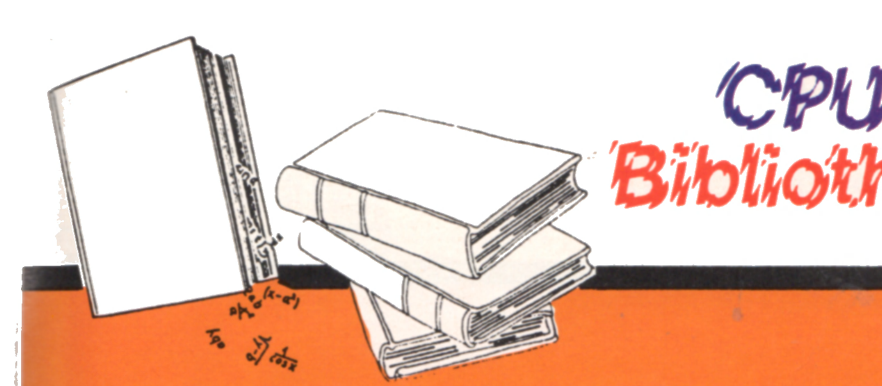
# Neuer EPSON GX-80-Drucker an Heimcomputer angepaßt

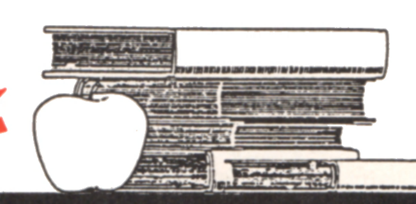
Mit dem neuen Matrixdrukker GX-80 liefert EPSON einen leistungsfähigen und preiswerten Drucker für den Heimcomputerbereich. Der GX-80 benötigt keine speziellen Interfaces mehr, sondern ist mit Hilfe von ROM-Modulen bereits fertig an den betreffenden Computer angepaßt. Bisher sind das Commodore 64 und Atari-Computer; andere werden in kurzen Abständen folgen. Der GX-80 wartet mit für den Heimbereich bemerkenswerten Eigenschaften auf: 100 Zeichen pro Sekunde, 1 KByte Buffer und Download-Bereich und neben dem normalen Schriftbild ein NLQ-(Letter-Quality-)Zeichensatz im ROM. Mit 32 internationalen Sonderzeichen deckt er die länderspezifischen Schriftarten ab, und eine Vielzahl programmierbarer Funktionen läßt eine vielseitige Verwendung zu. Der GX-80 transportiert Einzelblätter mittels Friktionsantrieb. Eine Kassette für automatischen Einzelblatteinzug sowie eine Traktorführung für Endlospapier sind optional erhältlich.

**Technische Daten des EPSON GX-80-Matrix-Drucker**

| | |
|---|---|
| Druckmethode: | Nadel-Matrix-Drucker |
| Geschwindigkeit: | 100 Zeichen pro Sekunde |
| Druckrichtung: | Bidirektional mit Druckwegoptimierung, unidirektional im Graphikmodus |
| Anzahl Nadeln: | 9 |
| Zeichensätze: | 2 x 96 ASCII-Zeichen mit je 32 internationalen Zeichen, desgl. NLQ-Zeichensatz im ROM |
| Schriftarten: | Pica und Elite normal, gesperrt, fett, komprimiert-gesperrt |

**Fortsetzung S. 78**






# Der Mikro-Computer im Beruf

**Heyne-Verlag, München; Autor: Manfred S. Schmidt; Preis 9,80 DM – Best.-Nr. 15/4**

Dieses Buch versucht die Materie „Computer" in der Form der Reportage an den Mann zu bringen: der Mikrocomputer in Büro, Kanzlei und Praxis, bei Freiberuflern, Selbständigen, Kleinunternehmern bis hin zu Managern und Veranstaltern von Autorennen.

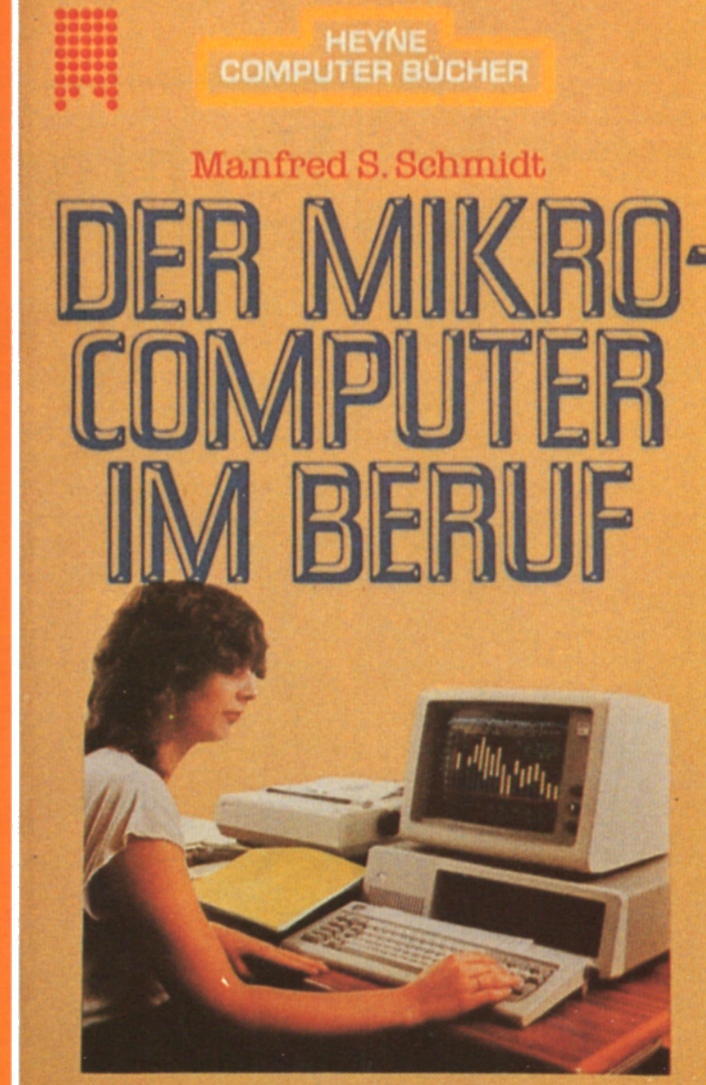
Der Autor dieses Buches, Manfred S. Schmidt, hat über zwei Jahre eine Vielzahl von Mikrocomputer-Anwendern besucht, um sie über ihre Erfahrungen, ihre Fehler und Erfolge zu befragen. Seine Berichte über die diversen Problemlösungen ergeben ein vielfältiges, lehrreiches Bild. Es kann dennoch nur eine Auswahl von Fallbeispielen sein, die keinen Anspruch auf Vollständigkeit erhebt, weder im Einzelnen noch im Spektrum aller möglichen Computeranwendungen.

Der Leser dieses Titels soll darüber informiert werden, welche Erfahrungen andere bereits mit dem Mikrocomputer gemacht haben, damit er selbst Fehlplanungen und Fehlnutzungen vermeidet. Und so bietet sich beim Lesen folgende Methode an: Suchen Sie sich anhand des Inhaltsverzeichnisses den Bereich heraus, dem Sie selbst zugehören, und gehen Sie anschließend zu verwandten Bereichen über – auch dort gibt es Anwendungen und Problemlösungen, die Sie interessieren werden.

**Einige Kapitel aus dem Inhaltsverzeichnis:**

Reportagen über den Einsatz von Mikrocomputern

1. Einzelhandel
2. Werbung – Direktverkauf – Versand – Großhandel – Transport
3. Vertretung – Vermittlung
4. Anwaltschaft – Kanzleiverwaltung
5. Handwerk
6. Landwirtschaft
7. Gastronomie
8. Baugewerbe
9. Architektur – Technik
10. Ärztliche Praxis – Medizinische Versorgung
11. Vereinswesen – Sport



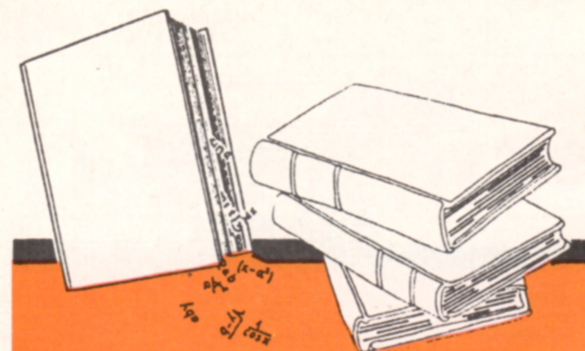

# Geldverdienen mit Programmierung

Dieses Buch wendet sich an die große Schar der Heimprogrammierer und Computerfreaks. Die Begeisterung dieser Leute ist dazu prädestiniert, einen wesentlichen Beitrag zur Überwindung der Softwarekrise zu leisten. Voraussetzung dazu ist, daß eine Brücke zwischen den Anforderungen der Industrie und den Produkten der Heimwerker geschaffen wird.

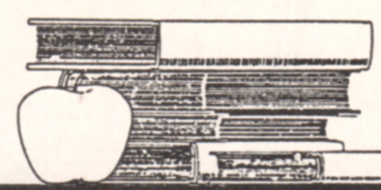
Das Ziel des Buches ist, hierzu einen wichtigen Beitrag zu leisten. Software hat heute immer dann eine große kommerzielle Chance, wenn sie profihaft erstellt und vermarktet wird. Was darunter zu verstehen ist, muß der Programmierer wissen, bevor er anfängt, seine kostbare Zeit in die Entwicklung von EDV-Systemen zu investieren.

Im ersten Kapitel des Buches werden Rahmenbedingungen für die Beschäftigung mit Software aufgezeigt. Dabei werden Antworten darauf gegeben, wie eine Programmieridee entsteht, ob ein fester Auftrag oder eine Eigenentwicklung interessanter ist, welche Voraussetzungen an Zeit, Raum, Geld und Know-how vorhanden sein müssen, und was an Aufgaben im

**(Fortsetzung nächste Seite)**



# CPU Bibliothek



Umfeld der Programmierung anfällt. Weiterhin wird auf den Markt und ein entsprechendes strategisches Vorgehen eingegangen.
Im zweiten Kapitel wird die Software-Entwicklung in vier Phasen beschrieben. Zunächst wird die Definition aus Anwendersicht erläutert. Grundsätzliche Designfragen sowie die Konstruktion des speziellen Softwarepaketes werden im Rahmen der zweiten Entwicklungsphase diskutiert. Die Programmierung als Kernpunkt der herkömmlichen Entwicklung wird kritisch beleuchtet. Im letzten Teil, der Test- und Einführungsphase, steht dann die Qualitäts- und Ablaufsicherung im Vordergrund. Zu allen Abschnitten des Entwicklungsprozesses werden Zielsetzungen, Anforderungen und generelle Vorgehensweisen dargestellt. Das Kapitel bildet damit einen Überblick über modernes Software Engineering.
Das dritte Kapitel ist der Vermarktung gewidmet. Vertriebsmöglichkeiten werden aufgezeigt, notwendige Werbe- und Marketingunterlagen dargestellt, Musterverträge und Dokumentationsanforderungen beschrieben. Eine größere Passage beschäftigt sich mit Verkaufstechnik. Schließlich wird auch noch die Verkaufsabwicklung und Kundenbetreuung angesprochen, wobei die Problematik der Begrenzung des Verwaltungsaufwandes im Vordergrund steht.
Im letzen Kapitel wird Bilanz gezogen und zwar in doppelter Weise. Die Überwachung der Softwareentwicklung nach betriebswirtschaftlichen und steuerlichen Gesichtspunkten bildet den offiziellen, gesetzlich notwendigen Teil. Hinzu kommt aber noch die sehr persönliche Antwort auf die Frage: hat es sich gelohnt.
In Anlagen finden Sie als Ergänzung Adressen und Literaturhinweise sowie die Aufforderung der Autoren an der nächsten Auflage durch eigene Beiträge und Erfahrungsberichte mitzuwirken.
Insgesamt enthält das Buch einen Überblick über die Software-Entwicklung und Vermarktung von der Programmidee über Realisierung und Verkauf bis hin zur Wartung. Die wichtigsten Grundregeln, Vorgehensweisen und Strategien werden mit banalen Tips und praktischen Anmerkungen zu einem Gesamtwerk gemischt, das in dieser Form noch nicht auf dem Markt existiert.

# Richtig Programmieren:

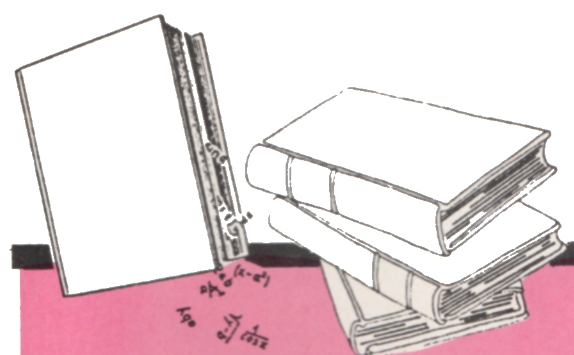
## Mit Basic zum Erfolg am Computer

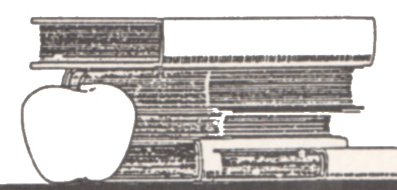
**Heyne-Verlag, München; Autor: Karl-Heinz Landeck; Preis 9,80 DM; Bestell-Nr. 15/10**

**Programmieren macht Spaß und bringt Gewinn!**
Jeder Anfänger kennt das: Die ersten Programmier-Versuche enden in Frust und Unlust. In diesem Buch wird den Profis über die Schulter geschaut und Schritt für Schritt vorgeführt:
- Wie professionelle Programme entwickelt und realisiert werden.
- Was der Einzelkämpfer am Heimcomputer daraus lernen und übernehmen kann.

Im Mittelpunkt steht BASIC. Eingebunden in kleine, leicht faßbare Problemstellungen werden die Möglichkeiten dieser populären Programmiersprache erschlossen – für Anfänger und Fortgeschrittene. Die wesentlichen Kommandos und Anweisungen werden gründlich erläutert.
Dieser Titel ist systemunabhängig verfaßt worden. Daher sollte es vom Leser, der bereits ein System zur Verfügung hat, immer im Zusammenhang mit dem Benutzerhandbuch für sein System und dem entsprechendem Basic-Handbuch benutzt werden. Der Autor hält sich hier, aufgrund der zahlreichen Basic-Dialekte, an das weitverbreitete Basic-80 von MICROSOFT.








# Computergestützter Fremdsprachenunterricht

**Herausgegeben von der Langenscheidt-Redaktion, Berlin/München.
Mit Beiträgen von 11 verschiedenen Autoren. ISBN 3-468-49434-3**

Ein Hauptproblem des computergestützten Unterrichts heute ist die unstrukturierte Art, wie Informationen über existierende und in der Entwicklung befindliche Computerprogramme verbreitet werden. Gute Programme – einschließlich solcher, die erst vor kurzem für Mikrocomputer entwickelt worden sind – wurden bisher nicht in einer Weise öffentlich dokumentiert, daß andere von ihnen lernen konnten. Nur wenige Programme sind im Handel erhältlich, und zahlreiche Lehrer/Programmierer arbeiten unabhängig voneinander und haben nur wenig oder gar keine Ahnung von existierenden oder in Entwicklung befindlichen Programmen. Bei Fortbildungsveranstaltungen ist es nicht selten, daß einfache Programme demonstriert werden, die weit hinter dem inzwischen erreichten Standard zurückbleiben.

**Aus dem Inhalt:**
Grundlagen der Arbeit mit Computern im Fremdsprachenunterricht – Anwendungsbeispiele aus den Bereichen Grammatik und Lexik und die Fertigkeiten Lesen, Schreiben, Hören und Sprechen – Möglichkeiten und Grenzen einer Übungstypologie.
Stand der Forschung in Europa und den USA – Hardware und Hardware-Entwicklung.
Erfahrungen mit Computern im Fremdsprachenunterricht an Universitäten und Schulen – Programme und Programmbeschreibungen.
Glossar zum computergestützten Fremdsprachenunterricht.

# MSX – Der erste Heimcomputer-Standard

## Einführung in: Technik, Programmierung und Anwendung

**Autor: Dietmar Eirich, Heyne-Verlag München – ISBN-3-453-47052-4. Preis: 12,80 DM**

Dieser Titel gibt Antwort auf folgende Fragen:
- Was heißt MSX – welche Technik steckt dahinter?
- Wie steht es mit der Software?
- Welche Anwendungen gibt es – welche sind wirklich sinnvoll?

Einige wichtige Kapitel aus dem Inhalt:
- Markt und Technik eines neuen Standards
- Firmen und Geräte: Hintergrundinformationen und Marktübersichten
- Peripherie und Zusatzgeräte
- Programmieren mit MSX-BASIC

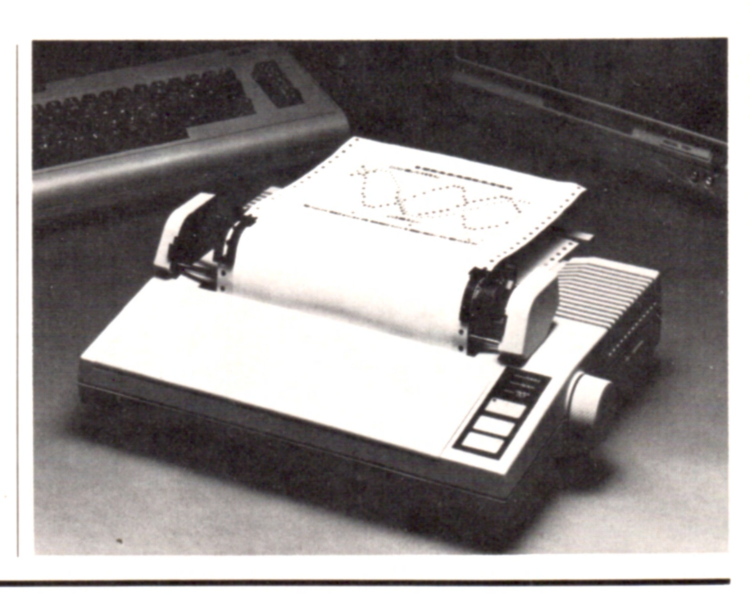
**Dieses Buch wurde bereits in der Juni-Ausgabe der „CPU" ausführlich vorgestellt.**



## Technische Daten des Epson GX-80-Matrix-Drucker

(Fortsetzung von Seite 74)

| | |
|---|---|
| Blattaufteilung: | Programmierbarer Zeilenvorschub, bis 32 Positionen hor. Tabulator 8 x 16 vert. Tabulatorpositionen |
| Papiertransport: | Walzenvorschub für Einzelblätter |
| Interfaces: | ROM-Cartridges für Commodore 64 und Atari, weitere in Vorbereitung |
| Input-Buffer und Download-Bereich: | 1 KByte |
| Optionen: | Traktorführung für Endlospapier, Einzelblatt-Kassetten-Aufsatz zum automatischen Einzug von Einzelblättern |



# EPSON QX-16: Mehrere Computer in einem

Der Personal Computer EPSON QX-16 erschließt eine Vielzahl von Anwendungen. Denn er ist als Dualprozessorsystem mit Zilog Z 80 und Intel 8088 aufgebaut. Somit stehen neben dem EPSON-spezifischen Betriebssystem auch CP/M 2.2, CCP/M und MS-DOS zur Verfügung, so daß der QX-16 auch IBM PC-kompatibel ist. Über ENABLE wird ein mitgeliefertes integriertes Programmpaket, das Dateiverwaltung, Kalkulation, bit-mapped Grafik und Telekommunikation umfaßt, bedient. Hinzu kommt Autopilot, mit dem Lösungen auch ohne Programmiererfahrung selbst geschneidert werden können. Je nach Betriebsart werden 2 x 320, 2x 360 oder 2 x 720 KByte auf Diskette gespeichert. Eine Festplattenversion kommt demnächst, der Controller ist schon integriert. 512 KByte RAM, 128 KByte Bildspeicher und hervorragende, vom Betriebssystem und dem Grafikprozessor NEC 7220 unterstützte Grafik erschließen neue Möglichkeiten in der Leistung von PCs. Der QX-16 ist IBM PC-kompatibel.




**Technische Daten des EPSON QX-16 Personal Computer:**

| | |
|---|---|
| Prozessoren: | Z 80 A, 8 bit, 4 MHz<br>8088, 16 bit, 5,3 MHz<br>Graphik NEC 7220, 17 MHz |
| Betriebssysteme: | EPSON CP/M 2.2 Multifont<br>Concurrent CP/M<br>MS-DOS, IBM PC-kompatibel |
| Arbeitsspeicher: | 512 KByte Ram |
| Massenspeicher: | 2 Floppy Disk Drives 5¼ Zoll<br>– 2 x 640 KByte im QX-16-Format<br>– 2 x 320 KByte im QX-10-Format<br>– 2 x 720 KByte im IBM PC-Format |
| Schnittstellen: | Seriell: RS 232 C (V.24), programmierbar<br>Parallel: Centronics 8-bit Standard |
| Steckplätze: | 3 |
| Bildschirm: | wahlweise<br>– 12 Zoll monochrom, hochauflösend 640 x 400 Punkte, bit-mapped, 25 Zeilen x 80 Zeichen, entspiegelt (serienmäßig)<br>– 14 Zoll Color, 640 x 400 Punkte (Zusatzkarte erforderlich)<br>– IBM-kompatibler Monitor, 640 x 200 Punkte |
| Tastatur: | Nach DIN, 105 Tasten mit Rechenblock, Cursorbkock, Funktionstastenblock, Cursorsteuerung auch mit standardmäßiger Maus |
| Mitgelieferte Software: | ENABLE:<br>Programmpaket aus 5 intigrierten Modulen mit Textverarbeitung, Kalkulation, Datenbank, Graphik, Kommunikation<br>SCOUT: Maus-Steuerung |

# Ihr Erfolgsprogramm – aus dem Tronic-Verlag




**Compute mit**
monatlich
Format A4
**2,80 DM**

**CPU**
monatlich
Format A4
**5,50 DM**

**Homecomputer**
zweimonatlich
Format A4
**6,– DM**

**Computronic**
zweimonatlich
Format A4
**6,50 DM**

**Spiele-Kassetten:**
Commodore 64
VC-20, Atari,
TI-99, ZX-Spectrum,
Laser 2001,
Schneider CPC 464
**Je 19,90 DM**

## Spitzenprodukte

**für den wachsenden Markt der Heimcomputer-Anwender**

**Informationen: Tronic-Verlag, Landstraße 29**

**Postfach 41, 3444 Wehretal 1, Ruf (0 56 51) 4 06 93**
**Vertrieb: Verlagsunion, Friedrich-Bergius-Straße 20, 6200 Wiesbaden, Ruf (0 61 21) 26 60**